PHD

Simulation optimisation: An expert mechanism approach

Mebrahtu, Habtom

Award date:
2006

Awarding institution:
University of Bath

SIMULATION OPTIMISATION : AN EXPERT MECHANISM APPROACH

Habtom Mebrahtu
Submitted for the degree of Doctor of Philosophy
University of Bath
Department of Mechanical Engineering

June 2006

AKNOWLEDGEMENTS

My deepest gratitude to my supervisor, **Professor Tony Mileham**
for his great help, encouragement, patience, comments and
for his fantastic direction throughout the research,
to make this research work possible.

I would also like to express my gratitude to **Dr Geraint Owen** whose advice and encouragement has been immense.

Special thanks to my colleague, **Rob Walker** whose unwaivered support, encouragement and discussions played a big role to the success of this research.

Special thanks to **SELEX S & AS**
particularly to **Mr Theodoros Dionysopoulos**
for providing me with industrial data and for their continuous support.

Above all I am deeply grateful to my **Yordanos Araya**
for her continuous support and encouragement, and
for putting up with me during difficult times.

Last, but not least,
thanks to my precious little daughters **Mary** and **Lydia**
without whose distraction, sought for attention and noise,
the research could have been completed much earlier.

...and anyone else I may have forgotten.

ABSTRACT

Manufacturing simulation involves taking input data, running of a system model and generating of outputs that are then used to assess performance and to help in decision making. The improvement of performance, however, has traditionally been undertaken using a trial-and-error approach. This approach is time consuming and, in many instances, does not guarantee an improved performance. There is a need for an optimisation system that could assist in assessing and enhancing the performance of simulation runs. This, together with a method for checking data consistency would be a valuable tool.

An Expert mechanism, integrated into a manufacturing simulator for the purpose of performance optimisation, has been developed. It assesses performance after each simulation run, then, using proven operations performance enhancing methods embedded in it, effects changes to the input variables with the aim of enhancing performance in the next simulation run. In contrast, existing commercial optimisation systems use metaheuristics in which a near-optimum value is searched from a population of alternative solutions. This is usually inefficient in terms of time and cost. To assist in achieving an effective optimisation process, the Expert mechanism also conducts an output data consistency check using the cusum chart.

The integrated simulator-Expert mechanism system is tested using three case studies. The results demonstrate that the expert mechanism is robust and an effective simulation optimisation system. The results are also compared with one of the most widely used commercial simulation optimisers to validate the system.

The research findings substantiated the research aims that a) an effective technique of performance optimisation could be coupled with manufacturing simulation to interpret simulation outputs, assess performance, and effect changes to model input variables for the purpose of performance enhancement; and b) a method could be derived to check simulation model data consistency because high output data variability should be avoided for effective performance optimisation.

The generic and flexible nature of the Expert mechanism makes it applicable to any discrete flow type manufacturing system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY AND LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AIT	Assembly/Inspection/Test
AOF	Average Objective Function value
Cooling schedule	The way temperature is decreased in simulated annealing
Energy landscape	Characteristics of the search space in simulated annealing
GA	Genetic Algorithm
MG	Maximum number of Generations
MPS	Master Production Schedule
RSM	Response Surface Methodologies
TOC	Theory Of Constraints
UAL	Upper Action Limit
WIP	Work In Progress
ACO	Ant Colony Optimisation
ARM	Adaptive Re-hope Method in particle swarm optimisation
BOF	Best Objective Function value
BOM	Bill of Materials
CR	Critical Ratio = time to due date/ process time remaining
CUSUM	Cumulative SUM chart
EDD	Earliest due date
EDM	Energetic Descent Method in particle swarm optimisation
EM	Expert Mechanism
Epoch length	No of iterations to make at a temperature in simulated annealing
FCFS	First Come First Serve
GNI	Number of generations with No significant Improvement
HC	Hill Climbing algorithm
JIT	Just In Time
LAL	Lower Action Limits
LDM	Lazy Descent Method in particle swarm optimisation
LEO	Linear move and Exchange move Optimisation
LIL	Local Iterative Levelling in particle swarm swarm optimisation
LP	Linear Programming
LPT	Longest Process Time first

LWL	Lower Warning Limit
MC	Maximum number of Configurations
MRP	Materials Requirement Planning
MTBF	Mean Time Between Failures
NP	Nested Partition method
OFP	Objective Function Precision
OLE	Object Linked Embedding
OPNDD	OPeration with Nearest operation Due Date first
OptQuest	A well known package for global optimisation
PI	Percent Improvement regarded as being significant improvement
PSO	Particle Swarm Optimisation
RTS	Reactive Thermostatistical Search
S/OPN	operation with lowest ratio of Slack time to Number of OPerations first
SA	Simulated Annealing
SELEX S & AS	SELEX Sensors & Airborne Systems
SET	minimum SETting time first
SLACK	operation with lowest SLACK time first
SPC	Statistical Process Control
SPT	Shortest Process Time first
SSM	Sequential Selection with Memory
TSP	Travelling Salesman Problem
UWL	Upper Warning Limit
VB	Visual Basic
VIMS	Visual Interactive Modelling Systems
V-mask	A method of checking cusum slope
Witness Optimizer	A commercial simulation optimiser from Lanner Group

CHAPTER 1
INTRODUCTION

This chapter gives a general overview of manufacturing systems and their process classification, introduces modelling and simulation, and broadly highlights the concept of performance optimisation using simulation. The proposed research aims are then stated.

1.1 Introduction to Manufacturing Systems

Manufacturing is the process of converting raw materials into products (Kalpakjian, 1995). It encompasses the design and manufacture of goods, using various production methods and techniques. A *system* is defined as a collection of entities, e.g., people or machines that act and interact together toward the accomplishment of some logical end (Law & Kelton, 2000A). Therefore, a *manufacturing system* can be described as a collection of entities that act and interact towards converting raw materials into products. A manufacturing system usually employs a series of value adding *manufacturing processes* to convert the raw materials into more useful forms and eventually into finished products (Wu, 1994A).

Manufacturing systems are dynamic and are in constant interaction with their environment at all stages of their activities – inputs, transformation processes and outputs. To help understand this complexity, the systems approach provides an overview which facilitates the analysis and understanding of complex subsystems.

Manufacturing System Classifications

The focus of operations management in manufacturing is on the transformation process which takes inputs and converts them to outputs, and the functions closely associated with this task (Hill, 2004). The transformation process and methods of conversion to produce goods are typically an interrelated set of processes feeding into one another as part of the total transformation.

There are several perspectives to be taken into account in understanding the choice of processing systems which can be made. This section outlines a brief account of classifications of the manufacturing system processes to illustrate some basic differences (Russel and Taylor, 2006; Hill, 2004; Heizer and Render, 2004A; and Petie, 1992). A summary of the system process types in respect of their layout, product volume, product variety, product positioning and implications are also shown in tables 1.1.1 and 1.1.2.

The **Project** process system is characterised by manufacturing of large-scale products which cannot be physically moved (Hill, 2004). It is identified by the large scale inputs in experience, know-how and skills coordinated to meet the individual needs of customers. The resource inputs will normally be taken to the point where the product is to be built. Typical examples are ship building, aerospace programmes and construction work.

Companies producing this type of product operate within markets that need a high degree of expertise, rapid product change and low sales volume. Manufacturing must meet this by having general purpose equipment with some specialist plant to meet particular project requirements. Processes will be highly flexible and able to cope with low production volumes of the market and the design changes that will occur during production.

The Jobbing process is chosen to meet one-off or small order requirements. The product tends to move around the resources i.e., it is smaller in size and generally simpler in nature than the case of the project process (Hill, 2004; Slack, 2004). More emphasis is placed upon the labour requirement. The process still needs to be flexible to meet the requirements of its customers, with major concern surrounding the utilisation of labour. Typical examples include production of a purpose built piece of equipment, Formula 1 parts, and special tools.

Batch production is characterised by increased volumes of jobbing production. The batch continuum starts at low volume in which capability is important with a high degree of product change and new product introduction (Russel and Taylor, 2006; Hill 2004). At high volume where price becomes more important, products become standardised, order sizes are increasing and product change is lower. This illustrates the shift in the production/marketing relationship towards the line process. The consequences on manufacturing are that a wide range of products and production volumes must be accommodated. This is generally associated with set ups between each batch quantity. Processes will be general purpose with a high degree of flexibility built in to meet the demand of the market. Utilisation of some parts of the plant may be low. Typical examples include moulding processes of different products where one mould to produce an item is put into a machine to make the item; then the mould is taken off, the raw material changed, another mould put into and another batch is made, and so on.

The **Line** process reflects the other end of the continuum to jobbing where high volume requirements justify dedicated repetitive processes to the needs of one or a small range of products. The businesses sell standard products which to be successful will be based typically on price and are associated with large customer orders (Hill, 2004). The level of product change will be restricted, and options may be supplied within strict guidelines. The process will be dedicated to the predetermined product range. High costs of change are associated with this type of process. Volumes will be high and need to be so in order to achieve the utilisation necessary to justify the immense capital outlay. Typical examples include production of white goods and the automotive industry.

The **Continuous** process is suitable for markets where product change and the rate of new product introduction are low. Companies will be selling product rather than capability, and orders will be won largely on price. The materials are transferred automatically from one part of the process to the next with the labour tasks being predominantly ones of system monitoring (Russel and Taylor, 2006; Hill, 2004). The manufacturing facilities to support the market will be low cost production. The process will be highly dedicated where the cost structure is based upon high production volumes. Typical examples include the petrochemicals industry and pharmaceutical industry.

When developing new products there has been a tradition of designing products on a sequential basis. This means that the right relationship between a product and its manufacturing process may not be aligned correctly, because the manufacturing aspect comes last.

PROCESS DESIGN	TYPICAL LAYOUT	TYPICAL PRODUCT POSITIONING
Flow Shop Continuous *Line (Dedicated Repetitive) *Batch or Intermittent	Line Product emphasis	Make-to-stock Assemble-to-order
Job Shop	Functional Process Emphasis	Make-to-order
Project/ Fixed Site	Fixed position Project emphasis	Make-to-order

Table 1.1.1. Traditional classification of System Design (Heizer & Render, 2005B).

	Project	Job Shop	Batch Flow	Line	Continuous
Volume	*Low*	*Low*	*Medium*	*High*	*High*
Variety	*High flexibility*	*High flexibility*	*Some flexibility*	*Low flexibility*	*Standardised*
Order winner	*High Quality*	*High Quality*	*High Quality*	*Competitive cost*	*Low cost*
Implication	*High cost*	*High cost*	*High cost*	*Some automation*	*Automated*
Machinery	*General purpose*	*General purpose*	*General purpose*	*Specific purpose*	*Specific purpose*
Product position	*Make-to-order*	*Make-to-order*	*Assemble-to-order*	*Make-to-stock*	*Make-to-stock*

Table 1.1.2. Product /Market/Process Characteristics (Slack et al, 2004).

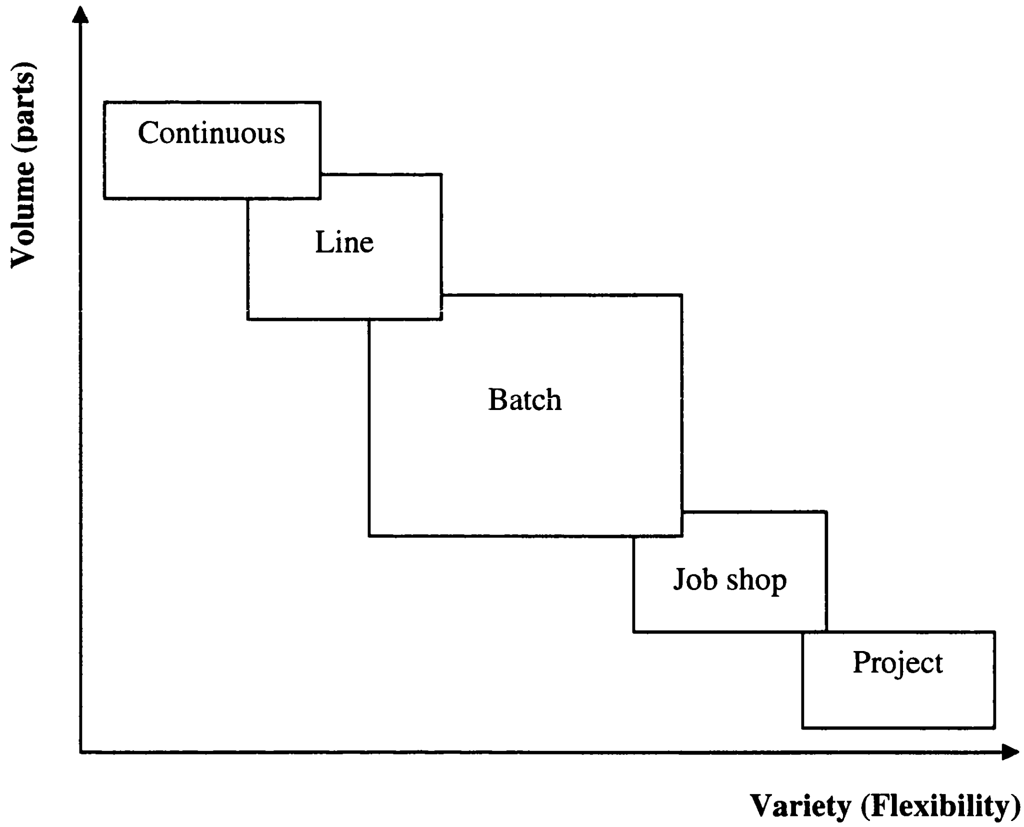


Figure 1.1 Classification of production process systems (Slack et al, 2004)

1.2 Introduction to Modelling and Simulation

1.2.1 Fundamentals of simulation

Experimentation with changes or alterations to existing physical manufacturing systems is rarely feasible because such an experiment would often be too costly or too disruptive to the system. In some cases the system may not even exist but nevertheless is wanted to be studied in its various proposed alternative configurations to see how it should be built in the first place. For these reasons it is usually necessary to build a *model* as a representation of the system and study it as a surrogate for the actual system. The majority of the models built for such purposes are mathematical models, representing a system in terms of logical and quantitative relationships that are then manipulated and changed to see how the model reacts, and thus how the system would react (Law and Kelton, 2000A).

Once the model has been built it must then be examined to see how it can be used to answer the questions of interest about the system it is supposed to represent. Simple models can be examined using analytical methods. However, many systems are highly complex, so that producing analytical solutions becomes difficult. In this case, the model must be studied by means of *simulation* (Law and Kelton, 2000A; Robinson, 2004A).

Developed in the 1950s, simulation is the process of building a model that mimics reality (Robinson, 2004A). A traditional definition of simulation (from Webster's Ninth New Collegiate Dictionary, Merriam-Webster, Inc, 1983) is;

> *The act or process of simulating; feigning; the imitative representation of the functioning of one system or process by means of the functions of another; examination of a problem often not subject to direct experimentation by means of a simulating device.*

Simulation software designers generally define simulation as;

> *Imitating the operations of various kinds of real-world facilities or processes, the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer.* (Lanner Group, 1998).

> *Experimentation with a simplified imitation (on a computer) of an operations system as it progresses through time, for the purpose of better understanding and/or improving that system (Robinson, 2004A).*

Simulation helps to monitor the dynamics of a system under various influencing conditions. It is the only appropriate analysis technique when formal mathematical methods cannot reflect the natural behaviour of a system (Law and Kelton, 2000A).

Law and Kelton (2000A) assert that in addition to the project credibility obtained by using simulation to visualise the system under investigation, the benefits of using simulation include;

- greater understanding of the system being studied,
- improved communication of ideas,
- reduced cost,
- ability to try many options quickly and easily.

Engineers and managers use simulation to improve the performance of existing manufacturing organisations, as well as to plan and design new systems. Simulation is a tool that has been widely accepted by managers for several reasons.. The main advantages of simulation, according to Heinze and Render (2005B), are as follows:

- Simulation is relatively straight forward and flexible.
- It can be used to analyse large and complex real-world situations that cannot be solved by conventional analytical models.
- Real world complications can be included that most operations management models cannot permit. For example, simulation can use any probability distribution the user defines; it does not require standard distributions.
- Time-compression is possible. The effect of process or design changes over many months or years can be obtained by computer simulation in a short time.
- Simulation allows "what-if" types of questions. Managers like to know in advance what options will be most attractive. With a computerised simulation model, a manager can try out several policy decisions within a matter of minutes.
- Simulation does not interfere with real-world systems. It may be too disruptive, for example, to experiment physically with new ideas in a manufacturing plant.
- Simulation can study the interactive effects of individual components or variables in order to determine which ones are important.

1.2.2 Types of Simulation

Simulation systems can be classified in many ways but among the useful ones are continuous versus discrete, static versus dynamic, and deterministic versus stochastic (Law and Kelton, 2000A; Robinson, 2004A).

Continuous vs. discrete simulation

Continuous simulation is used in conjunction with a system that can be represented by a series of flows and rates of flow. It is commonly associated with modelling continuous happenings of events where the state variables change continuously with respect to time, such as in the petroleum industry.

Discrete simulation deals with a system of discrete events for which the state variables change instantaneously at separated points in time, such as a manufacturing system with parts arriving and leaving at specific times. In fact, discrete event simulation is appropriate for the modelling of most manufacturing systems. A discrete event simulation is defined by Carson (1992) as;

> "*...one in which the system state variables change only at those discrete points in time at which events occur. Events occur from time to time as a consequence of activity times and delays.*"

The system state variables are the collection of variables needed to define the system state to a level of detail sufficient to meet the objectives of a particular project. The consequences of events could be;

- one or more states may change value;
- one or more conditional events may be triggered to occur, as a combined consequence of a current event and system state;
- one or more primary events may be scheduled to occur at some future simulation time.

Discrete event simulation is a way of representing what happens in reality by breaking it down into a calendar of events (Fernihough, 1997). The way the flow control of a discrete event simulation works is shown in Figure 1.2.2.

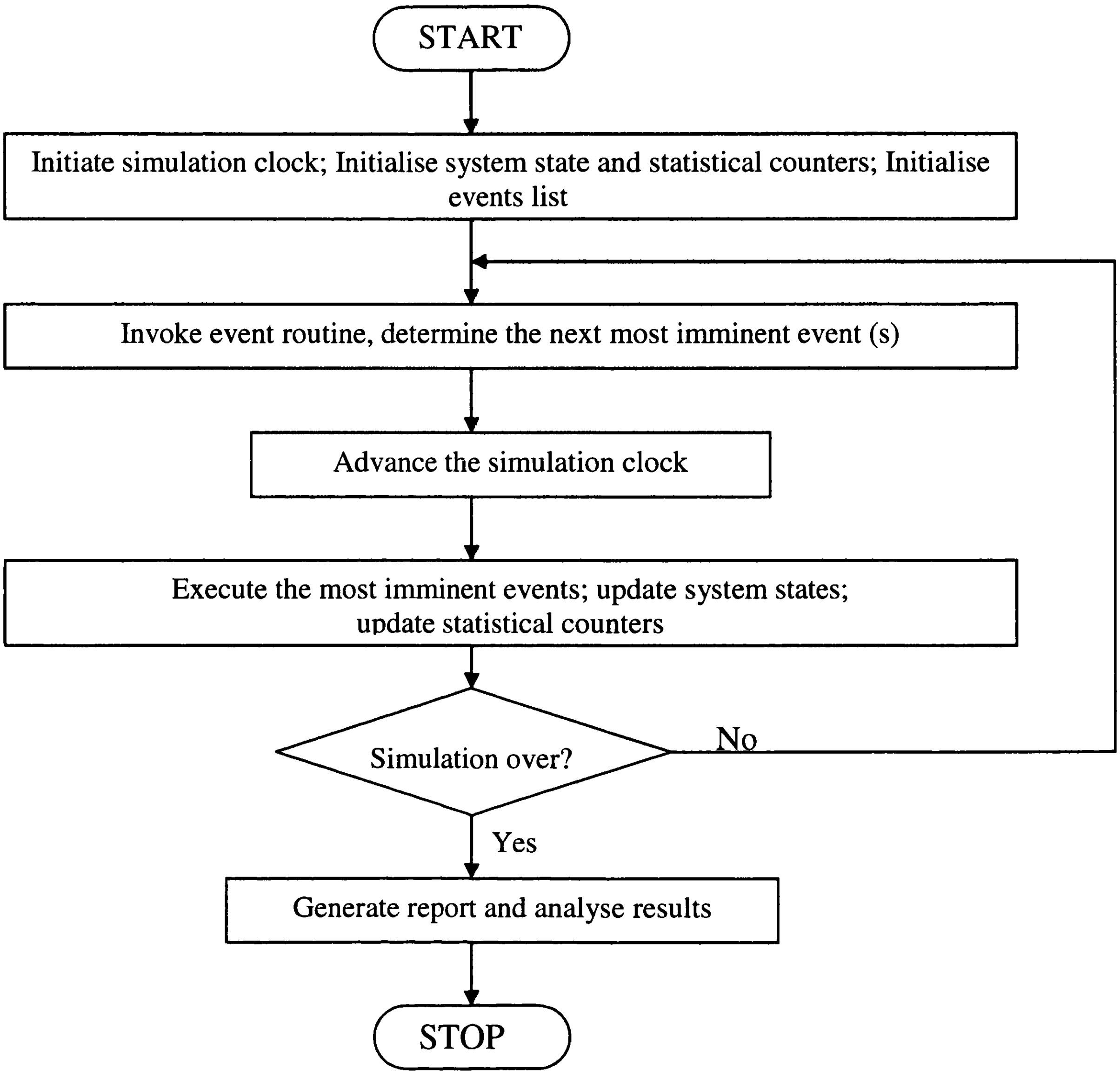


Figure 1.2.2 Flow control for the next-event based simulation (Fernihough, 1997)

Static vs. Dynamic Simulation models

A static simulation model is a representation of a system at a particular time, or one that may be used to represent a system in which time simply plays no role. A good example of static modelling is using a spreadsheet package for modelling materials requirement planning (MRP) systems.

A dynamic simulation model represents a system as it evolves over time.

Deterministic vs. Stochastic simulation models

A deterministic simulation model is one that does not contain probabilistic (random) components. A complicated and analytical intractable system of differential equations describing a chemical reaction might be a deterministic model. In deterministic models, the output is determined once the set of input quantities and relationships in the model have been specified even if it might take a lot of computer time to evaluate what it is.

Stochastic models operate with at least some random inputs. Most queuing and inventory systems are modelled stochastically. Stochastic simulation models produce output that is itself random, and must therefore be treated as only an estimate of the true characteristics of the model.

Most flow type manufacturing systems can be modelled by *discrete event* simulation as *dynamic* with *stochastic* elements.

1.2.3 Introduction to simulation optimisation

Manufacturing organisations are often faced with making tradeoffs in order to achieve desirable outcomes. Choosing these tradeoffs in the "best" way is the essence of the optimisation problem (Spall, 2003). A simulation model can be thought of as a mechanism that turns input parameters into output performance measures. In essence a simulation is just a function whose explicit form is unknown, since simulation is used instead of mathematical formulae where numbers could be plugged in.

One of the main goals of simulation is to find out how changes in the input parameters affect the output measures of performance. *Simulation optimisation* is a process of finding a combination of the input factors that optimises the output performance measure(s). Examples of performance measures include throughput, cost and work in progress. In general, the input factors in question could include discrete quantitative variables such as the number of machines at a work station in a manufacturing system, continuous quantitative variables such as the mean processing time for a machine, or qualitative variables such as the choice of a queue discipline. A detailed account on simulation optimisation is given at the later chapters.

1.3 Research Aims

Following a thorough search of academic and commercial literature (discussed in chapters 3 and 4), it is considered that a technique can be created that could assist production engineers and managers in interpreting results, and in assessing and optimising performance when using manufacturing simulation. The technique could include the use of proven performance enhancing methods that are not alien to manufacturing engineers and managers. There is also a need for a method to check data consistency of a manufacturing simulation model.

Consequently the overall aims of this research are

a) To conceive, develop and evaluate a system, integrated to a manufacturing simulator, that can assist in performance analysis and optimisation of a manufacturing system.

b) To derive a method to check simulation model data consistency and to evaluate solutions.

The main part of this research project concentrates on the mechanism that supports a manufacturing simulator in interpreting simulated results, assessing their performance and taking action to enhance performance. The powerful features of modern simulation packages are generally focussed on the front-end of building models easily and on getting simulation results quickly. As a result, massive reports and raw data are generated that do not typically help the user see the connection of these reports to the next appropriate action in a consistent and logical way. Additionally, limited alternative simulation models could be dealt with in a traditional "trail and error" way, but as alternatives increase, conducting a large number of simulation runs becomes time consuming and costly.

Some commercial simulation packages now include some types of integrated optimisation routines that generally work on a systematic search of optimum values from a pool of simulated results but do not make use of proven manufacturing engineering methods. Thus they are not usually user friendly to industrialists, require powerful computers and take a long time.

The proposed work is based on manufacturing engineering/operations management expertise embedded in the mechanism which is integrated to the manufacturing simulator. This method generally takes an action which is clearly understood by industrialists, and is more likely to enhance performance thus reducing the number of simulation runs required. In complex systems, simulation runs can take a long time.

The second part of this research deals with checking data consistency of the output data. For an effective performance improvement process it is imperative that the output data of a modelled system do not experience a high fluctuation. That is, the variability of the output data should be within an acceptable level. It is proposed that a method similar to that used for control charts could be used to check that the model output data are consistent and the solutions are dependable. This element could also be used to detect the significance of a change in performance to a given level. It would help to decide whether the changes made during the performance enhancement process have made significant improvement to the objective function or not.

1.4 Organisation of the Thesis

A general overview of manufacturing systems and their process classification has been discussed in *Chapter 1.* Modelling and simulation has been introduced and the concept of performance optimisation using simulation broadly highlighted. The proposed research aims are then stated.

Chapter 2 looks into manufacturing performance analysis methods. It particularly focuses on the most useful manufacturing performance evaluators and their relationships, followed by the most common analysis tools. The tools are then contrasted with simulation and the merits of manufacturing simulation highlighted.

Considering the need for advanced analysis tools in today's manufacturing environment, the use of simulation in manufacturing is covered in more detail in

Chapter 3. The structure and the main building elements of a typical manufacturing simulation model have been described. This chapter also discusses the limitations of current manufacturing simulation packages which have direct relevance to the research aims and objectives.

Chapter 4 reviews the literature on current work in manufacturing simulation, performance analysis, simulation optimisation and data consistency analysis. It examines the different types of optimisation techniques, from the classical approaches to the most recent, and reviews their applicability to simulation optimisation. It also discusses the concept of data consistency analysis and appropriate implementing methods. Finally it reviews the existing academic and commercial optimisation methods, and the consistency checking concept, and highlights their limitations and strengths.

Subsequently, *Chapter 5* re-states the research aims and describes the research objectives. The research methodology used to achieve the objectives is then discussed.

Consistent with the aims and objectives, the technical structure of the proposed system is discussed in *Chapter 6.* This chapter describes the simulator, the data consistency checking element and the optimisation element. It also depicts how these elements are integrated.

The rear-end elements of the simulator-optimiser system, the Expert mechanism, which is the main feature of the research, is discussed in detail in *Chapter 7.* The basic principles of integrating the Expert mechanism with the manufacturing simulator and the way the integrated system is developed are discussed in detail. This chapter also covers an overview of some proven manufacturing performance enhancing methods that are used in this research to demonstrate the concept.

Upon completion of the stages of developing the Expert mechanism and integrating it with the manufacturing simulator, *Chapter 8* uses a case study to demonstrate and, to an extent, validate the integrated system. *Chapter 9* continues with another case study whose objectives showed some addition to the first case study. This chapter also discusses the results of this case study that substantiated the effectiveness and robustness of the Expert mechanism.

A further case study from a world class company with real data and a validated simulation model is then covered in *Chapter 10.* This chapter, following a satisfactory set of results, discusses the success of the research. It also validates the system by comparing it with one of the leading commercial optimisation packages.

The implications of the research, its limitations and contributions, and the overall evaluation of the research findings are discussed in *Chapter 11.* Finally, *Chapter 12* draws the conclusions by discussing the findings of the research, and describes the further work recommended to expand and improve on the work covered in this thesis.

CHAPTER 2

MANUFACTURING PERFORMANCE ANALYSIS

This chapter looks into manufacturing performance analysis methods. It particularly focuses on the most useful manufacturing performance evaluators and their relationships, followed by the most common analysis tools in industry. The tools are then contrasted with simulation and the merits of manufacturing simulation highlighted.

2.1 Introduction

The pace of change in manufacturing is accelerating, product life cycles have been decreasing and companies are facing tougher competition. Also the complexity of modern manufacturing systems has put immense pressure on manufacturing firms. A manufacturing system can be assessed in many different ways depending on their goal but there is only one common aim – to make money (Goldratt and Cox, 2004; Browne et al, 1988). All activities in the business are but means of achieving this goal. From an operational point of view, three important criteria are useful in evaluating manufacturing progress towards making a profit. They are *throughput, inventory* and *operating costs*. Throughput is the rate at which saleable finished goods are generated. An increase in throughput has a direct impact on increasing profit. Inventory includes raw materials in stock, work in progress inventory and finished parts inventory. Reduction in inventory directly impacts return on investment and cash flow. Operating expenses are costs needed to convert inventory into throughput. They include costs of direct and indirect labour, heat, light, production facilities, etc. Understanding/analysing the factors that influence the attributes mentioned above is of paramount importance. Hans-Peter Wiendahl (1993) argues that the central building block of all concepts related to solutions for manufacturing is production planning and control. Among the logistic factors described include lead time, schedule observance and utilisation. Law and Kelton (2000B) identify throughput, lead time, work in progress (WIP) and bottleneck analysis as the main performance evaluators in manufacturing. A simplistic relationship between throughput, lead time and work in progress (assuming no scrap) is depicted in Figure 2.1

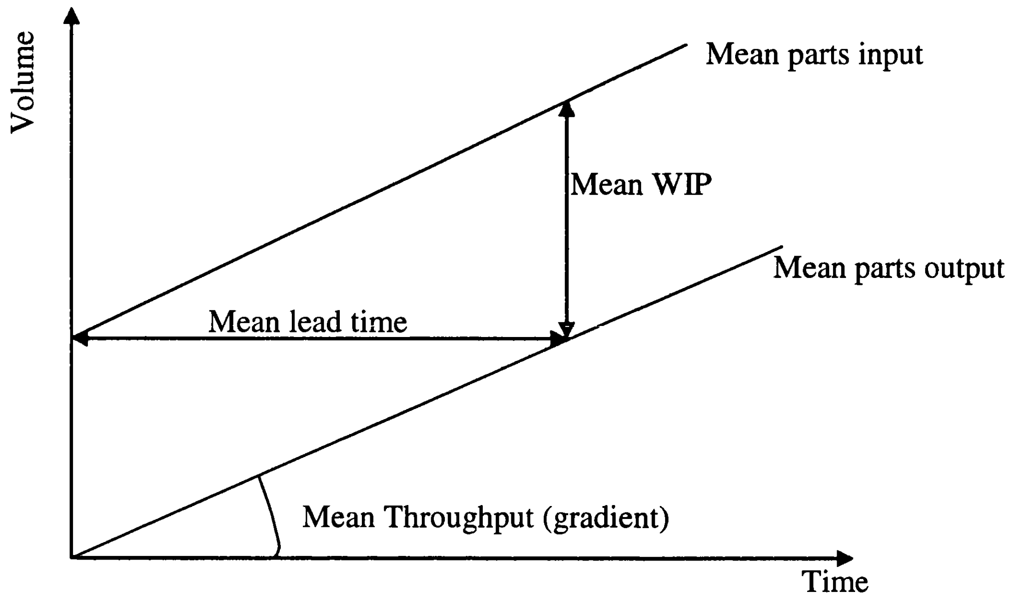


Figure 2.1 Throughput diagram (Steady state)

Some of the issues that are pertinent in manufacturing are:

- *the need for and the quantity of equipment and personnel* which includes number, type and layout of machines for a particular objective; requirements for material handling systems and other support equipment; location and size of buffers; effects of change in product volume and mix; effects of new equipment on an existing manufacturing line; labour requirements; and number of shifts.
- *Performance evaluation* that may be based on throughput analysis, lead time or bottleneck analysis.
- *Evaluation of operational procedures* which includes production scheduling (priority dispatching rules, batch sizes, routing, etc), policies of inventory levels, control strategies (conveyor system, AGVs, etc), reliability analysis (such as preventive maintenance), quality control policies, manufacturing strategies (MRP, JIT, etc).

The impact of changes in one or more of the factors affecting the above manufacturing issues on the overall system is of importance. Therefore, the need for advanced analysis tools in today's manufacturing is growing fast. Among the tools that are commonly used as decision support systems for shop floor performance analysis are back of the envelope calculation, spreadsheets, queuing theory and simulation (Baudin et al, 1992; Robinson, 2004A).

2.2 Back of the envelope calculations

Calculating orders of magnitude on the back of the envelope is widely used especially in small companies. It is useful to prove that such a calculation is either the maximum or minimum, depending on its application, but it may not give a realistic workable value where complex interaction of elements are involved. Referring to the throughput diagram shown in Figure 2.1, an example may look like the following:

If you release 5000 wafers every week into a process of 20 mask layers, then you will have to handle of the order of 100,000 photolithography operations every week which, on 24x7 mode, works out to 600 per hour. If each machine does 30 wafers/hour, then 20 machines will be needed.

This calculation can only prove that a capacity plan is infeasible but it cannot prove feasibility. If 25 machines are installed, there is no guarantee that 5000 wafers will be processed every week.

Mean lead time = Mean WIP/Throughout. Therefore if you allow 1000 lots of WIP in the shop floor and 200 finished parts come out every week, then the mean lead time will be 5 weeks. If the lead time is wanted to be 4 weeks then there must be around 800 lots as WIP.

Again here, the WIP may not be compatible with the throughput when constraints such as product mix are taken into account.

2.3 Spreadsheets

Spreadsheets, as perceived by most users, are natural extensions of back of the envelope calculations. Most spreadsheet packages do what normal back of the envelope calculations can do but provide easy ways to create, store, retrieve and perform calculations characterising relationships between process and production parameters. They also make it easy to assess the impact of changes by linking the data to organised tables and charts.

A typical application of a spreadsheet is in exploding a master production schedule (MPS) to subassemblies and components through a bill of materials (BOM) in a materials requirement planning (MRP) system. This is just an extension of an MRP system that can be completed manually. But by linking the MRP system with capacity, a useful static simulation can produce a chart that depicts a contrast of the production plan and available capacity.

The importance of a spreadsheet system cannot be under estimated but its application may have to involve correction (fudge) factors to compensate for errors that may arise due to interactions between elements and fluctuation of data. A good static simulator it may be, but it does not take the dynamics of manufacturing processes into account (Baudin et al, 1992; Robinson, 2004A).

2.4 Queuing theory

Queuing network – "flow-and-queue" – models, attempt to answer some the questions left open by spreadsheets. Baudin et al (1992) argue that queuing theory aim to predict the hitherto elusive WIP levels and cycle times, given a steady –state work load, process routes and simple resource allocation rules. Although this method has been tried by academics for analysing manufacturing systems, it has had limited success in industry. If this approach is to be made to work, it requires an excessive level of mathematical sophistication on the part of the user. Additionally, it is not established that a theory developed to model service systems is a good fit to manufacturing.

The above argument is supported by Robinson (2004A) who asserts on the use of queuing theory that a manager faced with a large set of mathematical equations may struggle to understand, or believe, the results from the model. One of the barriers has been a way to explain the approach without its mathematical underpinnings, or to communicate those effectively to manufacturing personel. Van Dijik (2000) also argues that, the application of queuing theory has remained rather restricted. The perception seems to have grown that queuing analysis is too detailed and too mathematically complex to allow for the direct practical application. Joustra et al (2001) consolidate Dijik's argument further by referring to the area where queuing theory is perceived to show some strength. They state that queuing theory is too restricted to predict and calculate queuing times even at check-in counters. The waiting formulas represent so called steady state situations. This would imply that the arrival rates of passengers are constant during long periods of time. This steady state assumption is clearly not the case with check-in arrival patterns. In contrast peakedness and variability is the major concern for planning.

2.5 Simulation

The move-queue-wait-seize-release-move ... approach of discrete-event simulation is similar to a manual board game where shop floor movements can be mimicked. It gives a reflection of what happens on the shop floor and helps to visualise the dynamics of the operations that are inherent in the system.

Unlike spreadsheets, simulation does not see the shop floor in terms of individual elements, but as a system as a whole, considering the relationships and interactions between different types of batches, resources, operators and work rules. Perhaps the greatest overall benefit of using simulation in a manufacturing environment is that it allows a manager or engineer to obtain a system-wide view of the effect of local changes to the manufacturing system (Law and Kelton, 2000B). Simulation models are dynamic and take into account dispatching rules, batching, priorities and shift effects. A comparison of the four analysis methods is shown in Figure 2.2.

In contrast to queuing theory, simulation can deal with the peaks and arrival patterns and give insight into short term effects, for example, half an hour peaks. In addition, simulation offers the freedom of using arbitrary distributions for the check-in processing time and arrival patterns (Joustra et al, 2001).

In comparison to direct experimentation, where the real system may have to be physically modelled, computer simulation has the advantages of lower cost, time savings, precise replication, no interruption and safety. A simulation model also has advantages over mathematical modelling which is usually unable to cope with dynamic and transient effects. Other operational research methods are also typically insufficient in aiding management decisions as conclusions may only be reached with too many constraints and assumptions (Wu, 1994B).

Until recently one of the main drawbacks of using simulation has been the need for high computer resource and the length of time needed to simulate complex systems. The current trend in the advent of very powerful computers at reasonable prices is overcoming this problem. There are very powerful and easy to use simulation packages on the market that include Witness (Lanner Group, www.lanner.com), ProModel (Promodel Corporations Inc, www.promodel.com), and Simul8 (by Visual Thinking International, www.simul8.com).

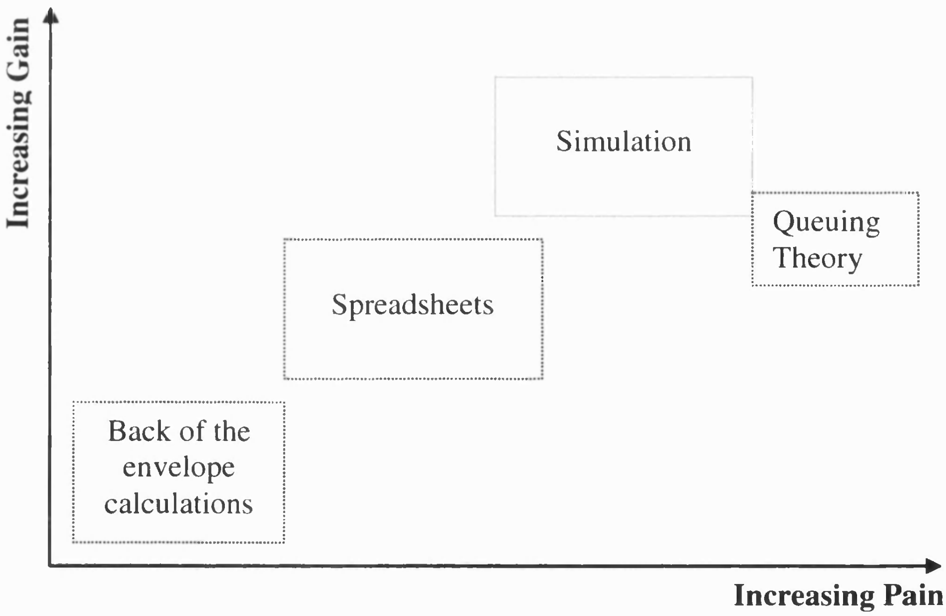


Figure 2.2 Comparison of common performance analysis systems (Baudin et al, 1992)

2.6 Simulation Based Manufacturing Analysis

The main steps in manufacturing simulation for the purpose of performance analysis can be summarised as follows (Heizer and Render, 2005B):

- defining the problem and identifying measure(s) of performance,
- acquiring relevant data such as details of resources, parts, working rules, etc,
- modelling the required system, simulating, validating and getting simulated results, and
- analysing the results and experimenting with model changes and assessing their effects.

Manufacturing simulation outputs are meant to provide information on how the simulated model performs. The output results may not contain data that are the exact match to the performance measure but may be the source data that could be used to assess performance. Typical shop floor performance measures include *throughput, operating cost, delivery performance, work in progress, resource utilisation* and *machine life*. Performance measures are chosen based on the investigation in question.

As previously discussed and also elaborated by Goldratt and Cox (2004), all activities in the business are but means to achieve the main goal – making profit. From an operational point of view, the three important criteria that are useful in evaluating manufacturing progress towards this goal are ***throughput, inventory*** and ***operating expenses***. Changes in any of these three elements such as increasing throughput or reducing the inventory level, result in changes in the bottom line. Similarly, Pegden et al (1990), mention the typical evaluation criteria in a manufacturing system's performance to be throughput and cost with additional concerns about delivery schedules, work in process and resource utilisation.

According to Law and Kelton (2000B), there are a number of potential benefits from using simulation in manufacturing analyses including

- increased throughout
- reduced work in progress inventories
- increased utilisation of machines and workers
- increased on-time deliveries of products to customers

- reduced operating expenses
- better understanding of the system prior to physically committing to changes, and
- ability to think about certain significant issues (such as system control logic) earlier in the design cycle.

Manufacturing simulation outputs include *final parts* as a direct match to throughput, average WIP as part of inventory, and element usage data as the means for assessing resource utilisation. These are just output results and the report does not give any indication of how these outputs are related and which factors affect their values. The general indication from simulation user companies is that they mainly depend on what is seen in the simulated results but have little knowledge of identifying the influencing factors to the chosen performance measures. Thus they use, usually a lengthy, trial-and-error method to obtain scenarios with better performances, which costs precious time. The following scenario would give an idea on generic manufacturing cases.

Taking cost as the main performance measure, requires initial knowledge of the costing parameters and the affecting factors. In addition to the data obtainable from simulation outputs (such as final parts, WIP, work time), the effects of each relevant input data and costing details need to be understood. So often, without computer assistance, even a simple model takes a long time to assess the cost performance. Then follows the questions " What should be done to reduce cost?", "Which simulation elements need to be examined?", "Which input data are relevant and how should they be amended?", and so on. A user with little or no deep knowledge of production engineering would have difficulties to answer these questions.

CHAPTER 3

MANUFACTURING SIMULATION

This chapter covers the use of simulation in manufacturing, in light of the need for advanced analysis tools in today's manufacturing environment. The structure and the main building elements of a typical manufacturing simulation model are described. It then discusses the limitations of current manufacturing simulation packages which have direct relevance to the research aims and objectives.

3.1 Introduction

The need for advanced analysis tools in today's manufacturing environment is increasing. This comes from the fact that manufacturing systems are growing more complex and integrated. The increased competition in many industries has resulted in a greater emphasis on automation to improve productivity and quality and also to reduce cost. Using simulation in manufacturing environment allows a system-wide view of the effect of "local" changes to the manufacturing system (Hurion, 1986; Carrie, 1998).

Generic manufacturing simulators normally contain the main representative elements of manufacturing industries such as machines, parts, labour, transport systems, etc.. Most manufacturing simulators on the market possess the ability of a system to be modelled with the help of some sub-models that have details of their own. Thomasma and Li-Xiang Yeo (1991) describe the human tendency to relate to complex systems by association, which means that the complexity is decomposed by first reorganising the discrete elements, or objects, of which the system consists. With the help of these sub-models or objects the user would be able to build a model more easily with proper relationships between them.

There are three main object groups involved in manufacturing simulation systems, as stated by Micheleti (1987):

- resource objects,
- entity objects, and

- requisite objects.

This is still true today. The "resource" objects include machines, operators, stores, and transportation units. The resource objects, when put together, sum up the physical limitation of the system. The resource objects are continuously present during a simulation run.

The "entity" objects represent the flowing entities (or transactions) in the system. These are typically parts and products flowing through the system. They are born (created) and may die (disposed) during a simulation run, and their life cycle in the system is of the main interest.

The "requisite" objects could be considered as a sub-group of the resource objects. They distinguish themselves from resources in that they are consumable. But they do indeed set up the limitations of the system in the same way resources do. Requisite objects are like machine lubricants, water, electricity, etc.

Each of these three groups represent an approach to system analysis by means of simulation. To get a complete comprehension of system dynamics, it is necessary to combine these objects.

3.2 Structure of Manufacturing Simulation Modelling

The essence or purpose of simulation modelling is to help the ultimate decision maker solve a problem. Therefore, good simulation modelling must merge good problem solving techniques with good software-engineering practice (Pegden et al, 1990). The following steps include the important stages of the simulation development process (Kelton et al, 2004B; Robinson, 2004B; Pegden et al, 1990)

1-Problem Definition: The simulation system has to be able to handle generic manufacturing problems. Although the layout models could be different for different manufacturing systems, there are common problems that need to be addressed such as

- What will be the throughput of the system? Will it meet production goals?
- Where are the bottlenecks? What can be changed to increase throughput?
- What is the best among several alternatives? How does the system performance change as a function of the number and type of machines, number of workers, types of automation, in- progress storage, etc.?
- What is the effect of a crucial machine breaking down?

The typical evaluation criteria are throughput and cost with additional concerns about delivery schedules, WIP, and resource utilisation.

The requirement, therefore, is to have a simulation system for decision support in solving problems similar to the ones mentioned above.

2-System Definition: In defining the system, it is useful to determine the boundaries and restrictions. Among the factors that need attention with this respect are;

- Computer Power: the system should consist of a reasonable number of components with as little detail and data as possible so that it can be run on available computers. This can be done by grouping similar components together and eliminating irrelevant ones. For instance, there is no need to develop objects such as lathe, grinder, etc.. All these can be grouped into one as a machine or station. Load/unload times can be included in process times unless they have clear significance on performance, and the effect of lubricants and electricity can be ignored especially if parameters like throughput are of importance.
- The system should be as easy as possible to model and should facilitate data entering by giving support to the user.
- The system will include components common to most manufacturing systems.
- It would be useful to include a mechanism that can help to add components to the ones built in the system. This can avoid some oversimplification that may result due to limited built-in components .

<u>*3-Conceptual Model Formulation*</u>: is developing a preliminary model(s) either graphically (such as flow chart) or in pseudo-code to define the ***components, descriptive variables*** and ***interactions*** that constitute the system.

It is desirable to design a model that neither oversimplifies the system to the point where the model becomes trivial (or worse, misleading) nor carries so much detail that it becomes clumsy and prohibitively demanding and expensive. A general overview of model complexity and the corresponding accuracy is depicted in Figure 3.2.1.

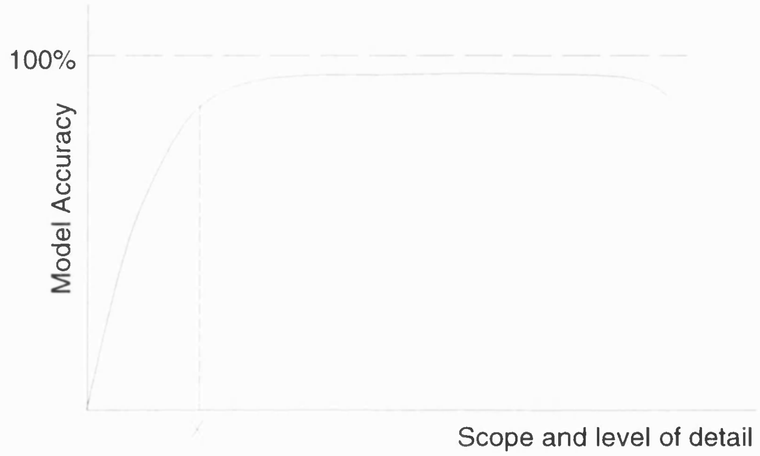


Figure 3.2.1 Simulation model complexity and accuracy (Robinson, 2004B)

The basic model elements include:

Components: The basic components that can represent most manufacturing systems are

- Resources: machines (processing units), stores or buffers, transport systems, operators;
- Transactions: Parts and products flowing through the system ;
- Queues: In-process storages for transactions;
- Operations: Things that happen to transactions to change their state;
- Arrivals: Raw material creation unit (or supply unit).
- Routing (or process plan)

- Additional elements such as failures and breakdowns.

Descriptive Variables: Each component is described by variables or attributes. The descriptive data of a component affects the interaction it will have with other components in the system. "Operation" for instance, can be described by: Name, processing time, resource(s) it seizes, set up time, tear down, input, output, efficiency, etc.. Similarly all the components have their own descriptive attributes.

Interactions: Once the components and variables are included in the system, the functional relationship among them should be determined. At this point the idea is to show the logic of the model i.e., "what happens". Usually graphics or block diagrams are used to describe the system. The interaction of the simulation components in the simulator is effected by an executable program that has got run-time parameters and the necessary execution procedures. This central part of the simulation system can be referred to as the KERNEL. The kernel may be written either in a simulation language or a general purpose language. It is this central part that combines the necessary parameters of the components and the required sequence of operations to effect simulation.

Having discussed the main fields in a manufacturing simulation, the next step would be to think of formulating the system in such a way that it would not be oversimplified or over-detailed.

Starting from the top, the system can be divided into three main model divisions:

. Simulation Kernel

. Layout modeller

. Entity flow modeller

These plus additional features are shown in Figure 3.2.2

Basically the two main modellers (layout and entity flow) need to be created before they can be used by the simulation kernel. One of the ways to facilitate the building up of the modellers is to create an object library whereby the simulation components are stored. Each component (or object) in turn would have a set of parameters (attributes) connected to it that describe and define the component. In many instances some details are fed to the model component by the user. Therefore, initially, the components would be built up to accommodate as much detail as may be needed. Once the building blocks have been formed then the model can be created by picking objects from the library and giving details as required. The objects may be designed to be a symbol (icon) which can easily be identified as a special construct.

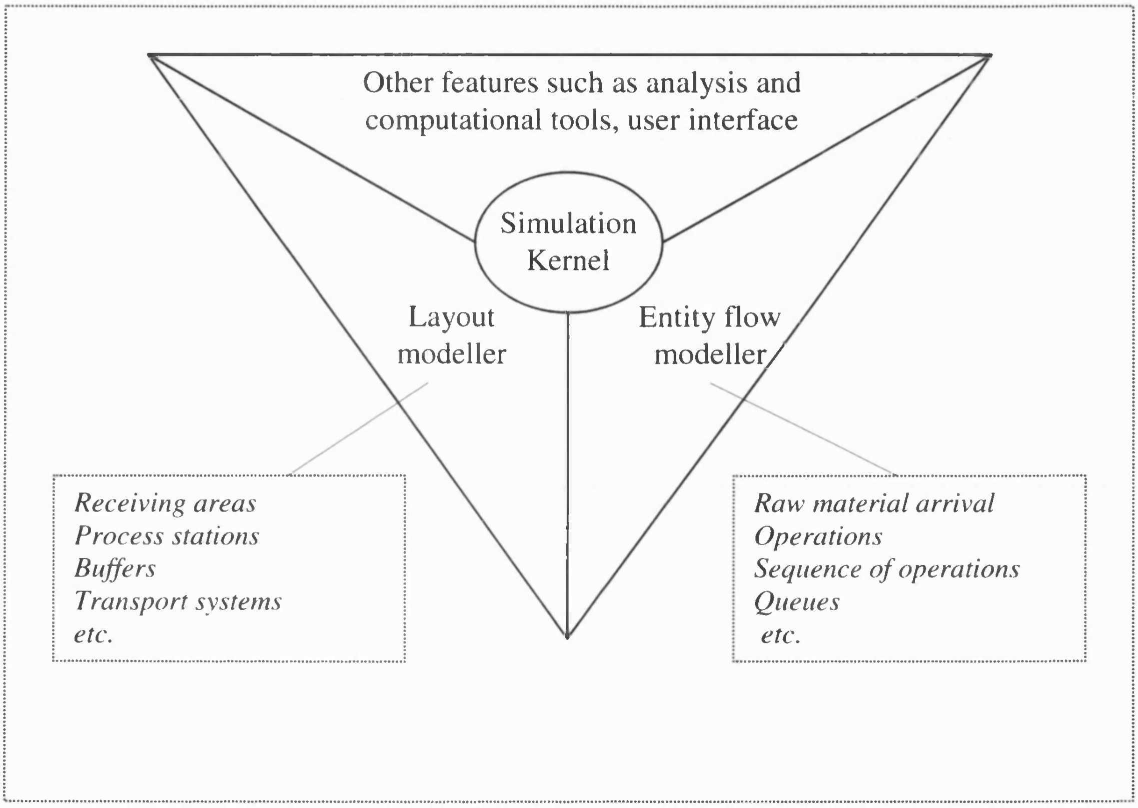


Figure 3.2.2 Manufacturing systems' model divisions.

Building simulation models can be further enhanced by introducing a mechanism to allow re-use of previously developed simulation models. According to Ulgen and Thomasma (1990), the idea is to use sub-systems from interconnected primitive elements (model components) and treat them as model elements i.e, subsystems are collections of other models but are treated as a single model. Icon-based simulation program generators that can manage sub-systems and allow libraries of simulation components encourage model reuse, thus decreasing the time required to build simulation models. The subsystem could allow the user to treat it like any other icon (copy, remove, move) and may have specialised operations such as Edit Subsystem and Show Details of Elements.

In addition to the ease of building models, if information on a group of manufacturing system components is not available, the entire group can be modelled as a simple workstation having average characteristics (such as processing time, downtime distribution) assigned to it. Later, if time, data availability, and the objectives of the study permit, this workstation can be replaced by a subsystem that models the group of components in greater detail.

4-Preliminary Experimental Design: includes selecting the measures of effectiveness to be used, the factors to be varied, and the levels of those factors to be investigated i.e, what data need to be gathered from the model, in what form, and to what extent.

The measures of effectiveness are parameters such as throughput, delivery date, bottleneck, WIP inventory, etc. The factors that affect these parameters need to be identified and the simulation outputs should contain data which would be used to evaluate them.

For instance, if "throughput" is the measure of effectiveness, then the factors that affect throughput could be the bottleneck, working hours of resources, number of resources, etc. And the simulation outputs important for this evaluation could be final parts, machine busy time, blockages, and buffer size.

Generally, in most manufacturing processes, final products produced, resource work time and work status, buffer capacity levels, transporter report status, raw material consumed, work in progress inventory and product make span are the needed simulation outputs. Production cost is also a vital output that may be added although it is usually complicated as it encompasses many details.

5-Input Data Preparation: Identifying the input data required by the generic model is also one of the main steps. Details of resources, transactions, operations, queues, arrivals and part routing (or process plan) are the input data. A good generic simulator will contain a systematic way of gathering the required information in a structured way.

Taking "operation" as an example, the input data may include Name of Operation, Processing time, Input part, Output part, Location of resource and Operators.

6-Model Translation: is formulating the model in an appropriate programming (simulation) language i.e. describing or programming the model in a language acceptable by computers. Here a decision will be required whether to use a general purpose language such as FORTRAN, PASCAL or C, or a pre-made simulation language such as GPSS, SIMSCRIPT, SIMAN and SLAM (Kelton et al , 2004A). In most instances, the latter is more appropriate as it reduces the development time considerably.

An alternative could be to use a high-level simulation package such as Witness (by Lanner Group), ProModel (by ProModel Corporation) or Arena (by Rockwell Software Corporation) which contains a library of most of the manufacturing elements.

7-Verification and Validation: The tasks and questions asked in this stage include: Does it work as expected (verification)? Verifying that the computer model represents the conceptual model faithfully; verifying that the expected things happen with obvious input, and walk through the logic with those familiar with the system. Can the generic system adequately represent the real-world system (validation)? Do the output performance measures from the model match up with those from reality? How is the confidence in the model's results? These questions are checked in the verification and validation stage. The verification and validation processes require reasonable experimentation with different scenarios.

Many authors have provided descriptions that outline the key processes in a simulation study. These include Law and Kelton (2000C), Banks et al (2001) and Robinson (2004C). A detailed inspection of the explanations show that they are in the main very similar, outlining a set of processes that must be performed. The main differences lie in the naming of the processes and the number of sub-processes into which they are split. The process involved can be summarised, as depicted by Robinson (2004C), Figure 3.2.3.

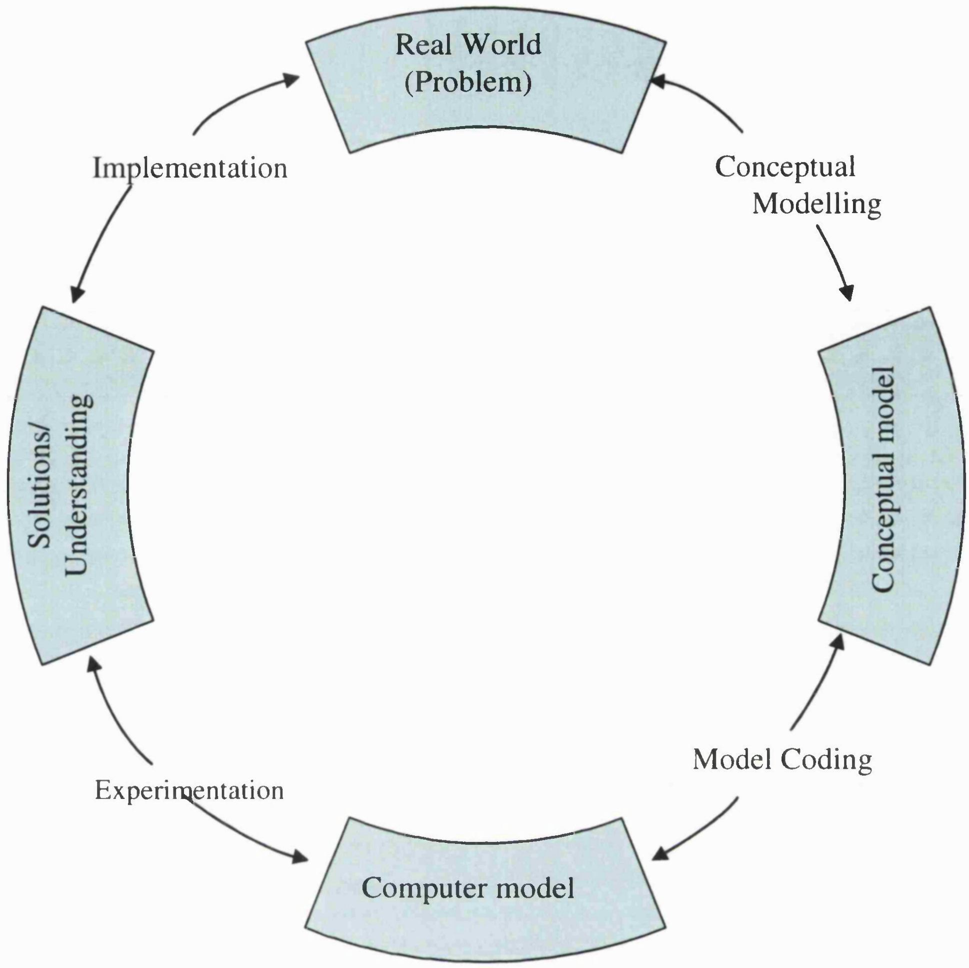


Figure 3.2.3 Simulation Studies (Robinson, 2004C)

3.3 Application of Simulation in Manufacturing

Although simulation in manufacturing has traditionally been used for high level capacity planning, there are many other benefits in using simulation. Factory layout, production routing, production mix, throughput prediction, bottleneck identification, new resources deployment, to name but a few, can all be predicted using simulation. In a modern manufacturing facility, the available flexibility introduces another degree of complexity in decision making (Law and Kelton, 2000B). The lack of a clear understanding of the dynamics and interaction of components of modern manufacturing systems calls for the use of simulation as an essential support tool. Simulation is no more a niche management tool, which can only be afforded by a few, thanks to ever increasing computer power and its affordable price (Heizer and Render, 2005B). The advancement in programming and software engineering also means that very clever simulation software has hit the market, with highly configurable user features and powerful animation.

Simulation is used for a wide range of applications in manufacturing. According to Robinson (1994), they could be summarised into eight categories:

- ***Facilities planning:*** When designing a new facility, simulation is used to ensure that it will perform correctly. By simulating the operations of the facility, bottlenecks are identified, shortages found and solutions sought.
- ***Obtaining best use of current facilities:*** Better performance solutions of current facilities could be identified but may be costly to implement and there may be uncertainty. The solutions can be tested at much lower risks with simulation.
- ***Developing Methods of Control:*** Experimenting with alternative control logic in a simulation enables the best practice to be identified. For instance, in a manufacturing plant a simulation model could be used to compare the effect of MRP II and KANBAN.
- ***Materials handling:*** A model of material flows and methods of handling would enable congestion points, shortages and weaknesses in control to be identified. Experiments could then be performed to improve the control and flow of materials.
- ***Examining the logistics of change:*** Simulation models are used to examine the logistics of change in order to minimise interruptions caused by changing existing facilities.
- ***Company modelling:*** Modelling of a company for simulating operations across more than one location can be undertaken. A high-level model could be created that includeds only essential details, and shows the flows of resources and information between sites. The interaction of the sites could be examined and experiments performed with alternative operating policies.
- ***Operational planning:*** Simulation is used in day-to-day planning and scheduling. A common application is to test a production schedule by simulating the resulting plant performance.
- ***Training operations staff:*** Simulation provides a low-risk environment in which supervisors and operators are trained in the operation of a facility.

3.4 Limitations of Current Simulation Packages

Although enormous investigation has been done on how to make manufacturing simulation easier to use and faster to get results, no significant effort has been made to improve the overwhelming demand at the back-end where a huge amount of results (in a form of tables and graphs) is pushed out for someone to make sense of it. A simulation can be thought of as a mechanism that turns input parameters into output performance measures (Law and Kelton, 2000A). In this sense a simulation is just a function, which may be vector valued or stochastic, the explicit form of which is also unknown.

One of the most popular model analyst companies, System Modelling Corporation (1992), stated that simulation packages cannot optimise a system's performance, nor can they solve problems, they can only describe the results of "what-if" questions.

Baudin et al (1992), while describing the advantages of using discrete-event simulation over spreadsheet and analytical queuing models, emphasised the substantial effort required for interpreting simulation results, which according to their experience, is a significant drawback of existing simulation packages.

The above assertions still hold with many simulation packages on the market. Massive reports and a lot of raw data are generated out of simulation runs but the user sees no connection of these reports to the next appropriate action in a consistent and logical way. Additionally, only limited alternative model solutions can be dealt with in a traditional way. As the possible alternatives increase, conducting a large number of simulation runs becomes time consuming and costly. Alam et al (2002) stated that with simulation modelling, the relationships between the design parameters and their resulting performance are not explicitly known. Therefore, simulation modelling becomes a trial-and-error process in which a set of input factors is used to generate output performance measures. The repetitive nature of this approach is often inefficient in terms of time and computing cost as well as in interpretation and prediction of results. Heizer and Render (2005B) also consolidate the fact that the trial-and-error approach in simulation does not generate an optimum solution. The conditions and constraints must be generated by users for solutions; the simulation model does not produce answers.

Cochran and Lin (1992), while explaining the need of meta-modelling for analysing the dynamic behaviour of assembly line systems, described the pitfall of computer simulation as producing only ad hoc results in the form of discrete data tables in which system parameters are not explicitly related. They added, the simulation results at most times are difficult to interpret with respect to overall system performance and can rarely provide a basis as a guide line for further improvement in system performance that eventually leads to optimisation.

CHAPTER 4
REVIEW OF CURRENT WORK

This chapter reviews the literature on current work in manufacturing simulation, performance analysis, simulation optimisation and data consistency analysis. It examines the different types of optimisation techniques, from the classical approaches to the most recent, and reviews their applicability to simulation optimisation. It also discusses the concept of data consistency analysis and appropriate implementing methods. Finally, it reviews the existing academic and commercial optimisation methods, and the consistency checking concept; and highlights their limitations and strengths.

4.1 Simulation and Performance Enhancement/Optimisation Methods in Manufacturing

4.1.1 Introduction

Simulation Optimisation means searching for the settings of controllable decision variables that yield the optimum expected performance of a system that is represented by a simulation model (Fu and Nelson 2003). Baesler et al (2002) define simulation optimisation as

> *"the combination of an optimisation method with a simulation model to determine the input variable settings that maximise the performance of the simulated system".*

Simulation optimisation is based on a process where the objective function, constraints or both are responses that can only reasonably be evaluated by computer simulation. Suppose a simulation model M, has n input variables ($x_1, x_2, \ldots x_n$) and m output variables ($y_1, y_2, \ldots, y_m$), the objective of the simulation optimisation is to find the optimum values ($x_1^*, x_2^*, \ldots, x_n^*$) for the input variables ($x_1, x_2, \ldots x_n$) that optimise the output variable(s). That is, it is the combination of an optimisation method with a simulation model to determine the input variable settings that maximise the performance of the simulated system (Azadivar 1992). In order to solve a simulation optimisation problem both the operational simulation model and an optimisation method or procedure are needed. To give an understanding of existing optimisation methods, some of the widely written optimisation techniques are discussed in the following sections.

4.1.2 Classical Simulation Optimisation Approaches

Fu (2002) and April et al (2003) identify the following classical approaches for optimising simulations:

- Ranking and selection.
- Stochastic approximation (gradient based approaches).
- Random search. and
- Sample path optimisation (also known as stochastic counter part).

Ranking and selection, unlike other optimisation procedures, evaluate exhaustively all the members from a given (fixed and finite) set of alternatives (Fu, 2000). Other common optimisation methods attempt to search efficiently through the given set to find improving solutions, because exhaustive search is impractical or impossible. Thus ranking and selection procedures focus on the comparison aspect, which is a statistical problem unique to stochastic setting. Two important concepts in the methodology have to do with the user specification of the indifference zone (level of precision) and the confidence level (probability of correct selection). This method is applicable mainly in cases of a low number of alternatives.

The stochastic approximation method dates back to over half a century. The algorithms attempt to mimic gradient search method used in deterministic optimisation. It works based on the gradient of the objective function in order to determine a search direction. Stochastic approximation targets continuous variable problems because of its close relationship with the steepest gradient search. Under appropriate conditions, one can guarantee convergence to the actual optimum with an infinite number of iterations. Unfortunately, because it is a gradient based method, stochastic approximation results in local optima, thus enhancements are required to find the global optimum.

Random search methods move through the solution space by randomly selecting points from the neighbourhood of a current point (Homem-De-Mello, 2003). Their advantage is their generality and the existence of theoretical convergence proofs. They have primarily been applied to discrete optimisation problems. A central part of the algorithm is defining an appropriate neighbourhood structure, which must be connected in a certain mathematical sense. The search algorithms move iteratively from a current single design point in the neighbourhood of the current point. Differences in algorithms manifest in two main fashions: a) how the next point is chosen, and b) the choice is usually between taking the current design point versus choosing the one that has been visited the most often. Unfortunately, April et al (2003) concluded that the theoretical convergence results mean little in practice where it is more important to find high quality solutions within a reasonable length of time than to guarantee convergence to the optimum in an infinite number of steps.

Sample path optimisation is a methodology that exploits the knowledge and experience developed for deterministic continuous optimisation problems. The idea is to optimise a function based on a relatively small single set sample using a number of sample paths (April et al, 2003). These sample paths are then averaged in an appropriate way, leading to an approximation to the objective function (Spall, 2003). A number of gradients needed in the search process are then computed based on this approximate function serving as a proxy for the true function. Because the sample path method is based on reusing the fixed set of simulation runs, the optimisation problem can effectively be treated as a deterministic optimisation problem using standard deterministic non-linear programming techniques. Generally, the sample size needs to be large for the approximating optimisation problem to be close to the original problem (Andradottir, 1998).

4.1.3 Response Surface Methodology (RSM)

In many publications, computer simulation optimisation has been linked to response surface methodologies (RSM) using parametric regression model approximations of simulation models. In other words, RSM is a numerical representation of a function that the simulation model represents (April et al, 2003; Nicolai et al 2004). A response surface is built by recording the responses obtained from running the simulation model over a list of specified values for the input factors. A response surface is in essence a plot that numerically characterises the unknown function. Hence, a response surface is not an algebraic representation of the unknown function.

RSM has been successful in performing sensitivity analysis within a limited parameter space and determining solutions satisfying constraints (Box and Draper, 1987). However, this parametric approach has not been successfully used to perform global approximations of a manufacturing simulation model because of its inability to provide a globally accurate fit to response functions.

Fu (2002) identified a case (SIMULA8'S OPTIMZ) where RSM has been used in a commercial package which was considered not to be efficient. Actually, according to April et al (2003), SIMULA8 has abandoned the use of OPTIMZ bringing the number of applications of this method to zero.

4.1.4 Simulation Meta-Models

To overcome the tedious iterative nature of the simulation modelling process, many researchers, including Kilmer et al (1993), Saad & Byrne (1995), Pierreval & Huntsinger (1992), Kleijnen (1979) and Alam et al (2002) proposed a metamodel. A metamodel is an auxiliary model that facilitates an understanding of the relationships between simulation inputs and outputs. Generally, simulation meta-models are organised in three layers- the input layer, a hidden layer and output layer. The hidden layer is a metamodel which is an algebraic model of the simulation. The metamodel approximates the response surface and therefore optimisers use it instead of the simulation model to estimate performance. Standard linear regression has been and continues to be one of the most used techniques to build metamodels in simulation. Generally, the metamodelling approach has mainly been applied to deterministic optimisation procedures to obtain an estimate of the optimum (Fu, 2002).

More recently, the hidden layer is commonly an artificial neural network, and is used to extract high-level features and to facilitate generalisation of outputs if the relationship between the input and output variables is non-linear (Laguna and Marti, 2002; Van Bears and Kleijnen, 2003; Kilmer and Smith, 1997). The input data are fed to the network at the input layer, and propagated with weights and activation functions to the output layer to provide responses. After presenting the sets of inputs and associated outputs, the network is able to learn the relationship between them by changing the weights of its connections. Then the variation between the network output vector and the known target vector is computed and back propagated through the artificial neural network. This is a process of training the neural network. Once the neural network is trained it would be able to compute output values from new input values.

Neural Network supported meta-models could prove very effective when the factors that influence a measure of performance in manufacturing systems are limited. Generally, the parameters to consider in manufacturing simulation processes are numerous and have complex relationships that the application of meta-models will be either long winded and very demanding in terms of computer power and time or limited in their accuracy. Cheng and Currie (2004) tried to overcome this problem by introducing a Bayesian approach within the metamodel but their approach does not go further than deterministic simulation optimisation.

4.1.5 Expert Systems

The use of expert systems in the performance enhancement of manufacturing systems along side simulation has been widely reported (Lyu & Gunasekaran,1997; Abdallah ,1994), Expert systems are computer applications that contain the knowledge, experience and judgment of skilled professionals. The performance level of an expert system is primarily a function of the size and quality of the knowledge base it acquired (Khorami, 1992). Waterman (1986) defines an expert system as;

The embodiment within a computer of a knowledge-based component from an expert skill in such a form that the machine can offer intelligent advice or make an intelligent decision about a processing function.

Lyu & Gunasekaran (1997), discuss the integration of an expert system with simulation to evaluate and improve scheduling strategies. They discuss a way of making the expert system powerful enough to effectively support the simulation system in solving problems. The entities in the simulation model were made to behave intelligently instead of acting simply based on some predetermined rules. Their framework had three parts - a simulation mechanism, an intelligent entities processor, and an expert system.

The simulation mechanism is the kernel of the model which controls the advance of time, future events lists and data collection. After users feed the simulation language system-related information, the simulation mechanism would take care of the rest of the tasks. The intelligent entities processor deals with the time update method, the generation of an appropriate event/process, system status update, process/event interaction, and statistics collection. The expert system consists of a knowledge base and inference engine and is there to help in displaying simulated results and analysing performance. As informative as Lyu & Gunasekaran's paper (1997) may be, it does not demonstrate an integrated link between a simulator and an expert system. This could be explained by the fact that they used SIMSCRIPT and may have required additional in-built subroutine adding capabilities which are currently available in many modern simulation systems on the market. Modern simulation packages like Witness have the functions of an "intelligent entities processor" in-built in them that, along with some logic capabilities, could perform the functions of the expert system described by Lyu & Gunasekaran.

Abdallah (1994) reported a similar knowledge based simulation model for job scheduling. He described the objectives of a knowledge base for a scheduling system as a mechanism to supply the decision maker (scheduler) with information that could be used to decide on different situations of the workshop such as increasing WIP; machine idleness; job due dates not satisfied; machine breakdown; unavailability of labour; rejection of certain operations or materials. The knowledge-base system would generate schedules and perform calculations on scheduling performance results such as job lateness, machine idleness, total work finished, percentage of work done, waiting time, etc. A design of experiments, based on simulation results, was then conducted as a single-factor scheme with two levels, and the results analysed. Scheduling rules such as shortest operation, longest operation, longest remaining time, next operation to total work ratio, earliest due date, operation slack, job slack and job slack ratio were used to generate the design of experiments. Having said that, the simulator was mainly used to test the results of a knowledge-based schedule generator rather than as a source to identify where attention should be given for enhanced performance.

Agarwal and Babu (1994) discuss the importance of the linkage of a simulator with other subroutines and modules for performance analysis. The subroutines and modules would be added to the front end of the simulator and are used to process input data to the simulator. The effect of different scenarios with variations in capacity, lot sizing and dispatching rules for shop floor performance were examined. Although the need for a support mechanism to the simulator was reported, the performance assessment was mainly done by displaying the simulation results in table and graph form, and did not use the simulation results as sources for improving performance.

Russell et al (1994), having explored works on the use of artificial intelligence (AI) to benefit simulation, assert the fact that the investigators agree on the need for some intelligent agent to control the behaviour of simulation. Their view is that the use of a single language to provide intelligent control to a simulation has some desirable aspects but may have two distinct drawbacks:

- it supposes that the language is capable of handling all the requirements of both the intelligent decision-maker and the simulation, and
- it presupposes that the programmer has expertise in both simulation and AI.

Their investigation supports the need to separately develop simulation from AI but later integrate them with a standard interface. Simulation languages provide elements that are specific to the construction of simulators: means to describe physical entities, attributes, and relations to other entities; mechanisms for specifying the passage of time; functions and procedures to gather and compute a variety of statistical measures; and means to provide for random events. Queuing structures are well-defined and typical simulation languages are capable of complex numeric calculations. In contrast, they argue, the strengths of AI lie in the ability to extract patterns from a complex set of facts, to reason on incomplete information, and to resolve conflicting goals. AI has the power of symbolic computing but is not as well suited for complex numeric computation.

Therefore, an ideal solution to intelligent simulation would take advantage of these language traits and strengths by keeping the physical simulation separate from the AI engine that makes decisions. The work of Russell et al (1994) presented a detailed analysis of the simulator-AI interface requirements and demonstrated their work with a case study on a small factory.

4.1.6 Metaheuristics

When dealing with the optimisation of complex systems, a course of action taken for many years has been to develop specialised heuristic procedures which, in general, do not require a mathematical formulation of the problem. Metaheuristics provide a way of considerably improving the performance of simple heuristic procedures (Brusha and Franek 2003; Laguna 1997). The search strategies proposed by metaheuristic methodologies result in interactive-procedures with the ability to escape local optimal points. *Genetic Algorithms (GAs)* and *scatter search* are metaheuristics designed to operate on a set of solutions that are maintained from iteration to iteration. On the other hand metaheuristics such as *simulated annealing* and *tabu search* typically maintain only one solution by applying mechanisms to change this solution from one iteration to the next. Generally, the most efficient procedures achieve their efficiencies by relying on the context information. The solution method can be viewed as the result of adapting metaheuristic strategies to specific optimisation problems. In these cases, there is no separation between the solution procedure and the model that represents the complex system.

4.1.7 Genetic Algorithms (GAs)

GAs comprise of a process of evolution of a large population of individuals (objects, chromosomes) before selecting optimal values. A GA designer provides an evaluation function, called fitness, that evaluates any individual. The fitter individual is given a greater chance to participate in forming of the new generation. Given an initial population of individuals, a GA proceeds by choosing individuals to be parents and then replacing members of the current population by the new individuals (offsprings) that are modified copies of their parents. The process of reproduction and replacement continues until a specified stop condition is satisfied or the predefined amount of time is exhausted. GAs use several genetic operators including selection, crossover and mutation (Baccouche et al 2003).

Autostat, a proprietary statistical analysis package available with AutoMod (a simulation package provided by AutoSimulations Inc, www.autosim.com) has an optimisation element that incorporates an evolutionary strategies algorithm which is a variation of genetic algorithm (Fu, 2002; Law and Kelton, 2000D). The optimisation module handles multiple objectives by requiring weights to form a fitness function. The user selects the input variables (factors) to optimise and the performance measures (responses) of interest. For each input variable, the user specifies a range of values, and for each performance measure, the user specifies the relative importance. The termination condition is also specified by the user. The total number of generations is set at seven times the number of parents per generation, the latter of which is also user specified. Initially, at what is called generation 0, the package randomly generates 21 model configurations and simulates them. The three configurations with the best objective-function values are selected for generation 1. At generation 1, these three configurations are used to generate 21 new configurations randomly, which are then simulated. The three best of these new configurations are selected for generation 2, etc. A given configuration is never simulated more than once, so that some configurations in a generation may not have to be newly simulated. The stopping rule has three user-specified parameters: the Maximum number of Generations (MG), the number of Generations with No significant Improvement (GNI), and the Percent Improvement regarded as being significant improvement (PI). Suppose that

MG=50, GNI=5, and PI =5

and the algorithm has not terminated at generations $j, j+1, \ldots, j+4$:
Let BOF(j) be the Best Objective Function value for the 21 system configurations at generation j. Then, the algorithm will terminate at generation J + 5 if

$$|BOF(j+5) - BOF(j)| / |BOF(j)| \leq 0.05 \qquad (4.1)$$

otherwise, it will go on to generation j + 6. In this case the algorithm will terminate between generations 6 and 50.

When the optimisation process is complete, the top 30 configurations are displayed, along with various summary statistics from the simulation replications.

SimRunner, another proprietary optimisation package linked to the ProModel simulation platform (Law and Kelton, 2000D) is also based on evolutionary strategies initialised by genetic algorithm generation. The stopping rule depends on "Optimisation Profile" and "Objective Function Precision" which are both user-specified. The options for optimisation Profile are "Aggressive", "Moderate" and "Cautious", corresponding to three different and increasing values of an internally determined population size (PS). As PS is increased, the algorithm generally runs for more configurations and identifies better ones. Objective Function Precision (OFP) is a real number used to decide when to terminate. If BOF and AOF are the Best and Average Objective Function values respectively, for the PS configurations in a particular generation, the algorithm terminates at this generation if

$$|\mathrm{BOF} - \mathrm{AOF}| \leq \mathrm{OFP} \tag{4.2}$$

otherwise, the next generation of PS system configurations will be selected and simulated, BOF and AOF recomputed, and the test done again, etc.

4.1.8 Simulated Annealing (SA)

Barretto et al (1999), while discussing the Linear move and Exchange move Optimisation (LEO) as applied to simulation optimisation, describe simulated annealing (SA) as a method based on Monte Carlo simulation, which solves difficult combinatorial optimisation problems. The name comes from the behaviour of physical systems when melting a substance and lowering its temperature slowly until it reaches freezing point. Simulated annealing was first used for optimisation by Kirkpatrick et al (1983).

A simulated annealing optimisation starts with a Metropolis Monte Carlo simulation at a high temperature (Luke, 2003). This means that a relatively large percentage of the random steps that result in an increase in energy will be accepted. After a sufficient number of Monte Carlo steps, or attempts, the temperature is decreased. The Metropolis Monte Carlo simulation is then continued. This process is repeated until the final temperature is reached. A Simulated Annealing program consists of a pair of nested DO-loops. The outer-most loop sets the temperature and the inner-most loop runs a Metropolis Monte Carlo simulation at that temperature. The way in which the temperature is decreased is known as the *cooling schedule*. In practice, two different cooling schedules are predominantly used; a linear cooling schedule ($T_{new} = T_{old} - dT$) and a proportional cooling schedule ($T_{new}=R \times T_{old}$) where $R<1.0$. These are not the only possible cooling schedules; they are just the ones that appear the most in the literature.

The more difficult aspect is to make a decision on determining the cooling schedule (initial temperature and how the temperature is lowered) and to determine how many iterations to make at each temperature (Epoch length). The epoch length depends upon the maximum size of the Monte Carlo step at each temperature. While a pure Metropolis Monte Carlo simulation attempts to reproduce the correct Boltzmann distribution at a given temperature, the inner-loop of a simulated annealing optimisation only needs to be run long enough to explore the regions of search space that should be reasonably populated. This allows for a reduction in the number of Monte Carlo steps at each temperature, but the balance between the maximum step size and the number of Monte Carlo steps is often difficult to achieve, and depends very much on the characteristics of the search space or *energy landscape*.

Suppose there is a solution space S and an objective function C (real function), the purpose is to find a solution (or state) $i \in S$ that optimises C over S. SA makes use of an iterative improvement procedure which is determined by a neighbourhood generation. So starting with an initial state, a neighbour state is generated, and the algorithm either accepts or rejects this based on fitness improvement in conjunction with the stochastic probability of $e^{-\delta/t}$ (Metropolis criterion). δ is the difference in fitness between the current state and its neighbour, and t is the current temperature (control parameter). t is a mechanism to avoid local optima. The higher the temperature, the higher the probability of acceptance will be. SA works by means of searching and evaluating a set of feasible solutions, reducing the possibility of becoming trapped in a local optima at the early stages of search.

The general SA algorithm in pseudo code, with the adopted cooling schedule is as follows (Barretto et al 1999, Anderson 2001):

```
Initialise t to hot   {initial state}
Repeat
Repeat
k:=0;
Generate state j        {neighbour hood state}
δ= C(j) – C(i);         {fitness improvement]
If δ>0  then i:=j
else if random (0,1)> e^(-δ/t) then   i:=j;
k:=k+1;
until k=n;
t:=R.t;            {reduce t slowly}
until t<t_f
```

The probability of acceptance as defined by Anderson is P[accept C(j)] = $1/\{1+ e^{-\delta/T}\}$.

SA is generally applied to single objective optimisation problems but Avello et al (2004) tried to explore ways of using SA in cases of multi-objective optimisation problems. Their approach included the introduction of weighting to each objective function and allocating separate nested cooling curves. In the case where the trend of changes in the objective function values vary, one objective, known as the reference objective, leads the search. The weightings are selected based on random selection, current performance and historical performance. Unfortunately, this concept is put only as a theory and has neither been proven by a case study nor presented with a demonstration to give confidence in its industrial application.

Simulated annealing has been commercially applied in Optimiser within the Witness simulation platform of Lanner Group (2004). The primary search strategies in optimiser use simulated annealing and tabu search. Optimiser being a proprietary software, the author has not thus far obtained detailed account of how these search strategies are structured but the basic termination framework is as follows. Optimiser works on two user-specified parameters; the maximum number of configurations (MC) and the number of configurations for which there is no improvement (GNI) in the value of the objective function. Suppose

MC = 200,   GNI = 15

And the value of the objective function at configuration j is the best up to that point. Then the algorithm will terminate at configuration J + 15 if none of the objective function values at configurations J + 1, J +2, ..., J+15 is better than at configuration j. However, the algorithm will never go beyond 200 configurations.

Debuse et al (1999) briefly described the incorporation of reactive thermostatistical search (RTS) within the Witness discrete event simulation package. RTS is a technique, based upon simulated annealing, which extends the concept of thermostatistical persistency which aims to make obvious decisions in the search process early on so that more effort may be concentrated on the more difficult choices. The algorithm includes elements of tabu search by learning from its experience of the problem domain and modifying its search strategy accordingly. RTS monitors the performance of each of the simulation parameters and adapts them accordingly. The search gives bias towards simulated parameters which when modified have recently given solutions which are accepted as replacements for the current solution, using an adaptive neighbourhood. Rather than selecting a simulation parameter at random to modify, in the generation of a neighbourhood, this technique selects each parameter with a probability based upon its past performance.

4.1.9 Tabu Search

Tabu search is a metaheuristic that guides a local heuristic search strategy to explore the solution space beyond local optimality (Dengiz and Alabas, 2000). Tabu search operates by identifying key attributes of moves or solutions in the neighbourhood, and imposing restrictions on subsets of these attributes, depending on the search history. The two prominent ways of exploiting search history in tabu search are through *recency* memory and *frequency* memory. Recency memory is typically a short-term memory that is managed by structures or arrays called "tabu lists", while frequency memory more usually fulfils a long term search function (Glover et al, 1999). A standard form of recency memory discourages moves that lead to solutions recently visited. A standard form of frequency memory discourages moves leading to solutions whose attributes have often been shared by solutions visited during the search, or alternatively encourages moves leading to solutions whose attributes have rarely been seen before. Another standard form of frequency memory is defined over subsets of elite solutions to fulfil an intensification function that reinforces the inclusion of special attributes of these solutions within new solutions.

Short and long term components based on recency and frequency memory are used separately and together in complementary tabu search strategies. This approach operates by simply modifying the neighbourhood of the current solution. The introduction and exploitation of these adaptive memory strategies within tabu search distinguishes it from other metaheuristic approaches, and endows it with an ability to learn how to make its way effectively through solution spaces.

The combination of the power of tabu search with a complementary population-based approach (such as scatter search) can yield a method of remarkable power for problems that unite the concerns of simulation and optimisation.

4.1.9 Scatter Search

Scatter search is a methodology that operates on a population of solution vectors X. Laguna (1997) describe the procedure of scatter search as:

1) Apply heuristic processes to generate a starting set of solution vectors (trial points). Designate a subset of the best vectors to be reference points.
2) Form linear combinations of subsets of the current reference points to create new points. The linear combinations are i) chosen to produce points both inside and outside the convex region spanned by the reference points, ii) modified by generalised rounding processes to yield integer values for integer-constrained vector components.
3) Extract a collection of the best points generated in step 2 to be used as starting points for a new application of the heuristic processes of step 1.

Repeat these steps until reaching a specified iteration limit. (Subsequent iterations of step 1 incorporate advanced starting solutions and best solutions from previous history as candidates for the reference points.)

The main application of scatter search and tabu search has been in one of the most commercially applied optimisation packages, OptQuest.

4.1.11 Scatter Search in OptQuest

OptQuest is a well known package for global optimisation (Jones and White, 2004; Schwetman, 2000) and employs a combination of scatter search and tabu search. Scatter search, as stated above, operates on a population of controls to determine the next control evaluation. This control is generated by a linear combination of the reference controls mapped over the feasible region. Tabu search is superimposed over the process to prevent exploring regions of the response population that have been previously probed.

According to Glover and Laguna (1997), OptQuest seeks to find an optimal solution to a problem defined on a set X of bounded variables. The scatter search procedure in OptQuest starts by generating an initial population of reference points (population of X vectors). The initial population may include an initial point suggested by the user, and it always includes the midpoint

$$x_i = l_i + (u_i - l_i)/2 \qquad \text{for } i = 1, 2, \ldots, n \qquad (4.3)$$

where $L = \{ l_i : i=1,2,\ldots n\}$ is the set of lower bound values and $U = \{ u_i : i=1,2,\ldots,n\}$ is the upper bound values for all $x_i \varepsilon X$ (Laguna 1997). Additional points are generated with the goal of creating a diverse population. A population is considered diverse if its elements are significantly different from one another. The system uses a Euclidean distance measure to determine how close a potential new point is from the points already in the population, in order to decide whether the point is included or discarded. Since the system allows for linear constraints to be imposed on a solution X, newly created reference points for X are subjected to a feasibility test before they are evaluated (i.e, before the objective function value f(X) is calculated). Evaluation of the objective function entails the execution of a simulation model. The set of constraints are generally represented as $AX < B$. The feasibility test consists of checking (one by one) whether the linear constraints imposed by the user are satisfied. An infeasible point X is made feasible by formulating and solving a linear programming (LP) problem. The LP has the goal of finding a feasible X* that minimises the absolute deviation between X and X*. Mathematically, Minimise $d^- + d^+$

Subject to $AX^* \leq B$

$X - X^* + d^- + d^+ = 0$ and

$L \leq X^* \leq U$

Where d^- and d^+ are negative and positive deviations from the feasible point X^* to the infeasible reference point X. When constraints are not specified, infeasible points are made feasible by simply adjusting variable values to their closest bound. That is, if $x_i > u_i$, then $x_i^* = u_i$, for all i = 1,2,...,n. Similarly if $x_i < l_i$, then $x_i^* = l_i$, for all i = 1,2,...,n.

The population size is automatically adjusted by the system considering the time that is required to complete one evaluation of f(X) and the time limit the user has allowed the system to search. Once the population is generated, the procedure iterates in search of improved outcomes. At each iteration two reference points are selected to create four offsprings. Let the parent reference points be X_1 and X_2, then the offsprings X_3 to X_6 are created as follows:

$$X_3 = X_1 + d \tag{4.4}$$

$$X_4 = X_1 - d \tag{4.5}$$

$$X_5 = X_2 + d \tag{4.6}$$

$$X_6 = X_2 - d \tag{4.7}$$

Where $d = (X_1 - X_2)/3$. The section of X_1 and X_2 is biased by the values $f(X_1)$ and $f(X_2)$ as well as the search history. In particular, tabu search memory functions are used to keep track of those reference points that have been recently used to create linear combinations. An iteration ends by replacing the worst parent with the best offspring, and giving the surviving parent a tabu-active status for a given number of iterations (i.e, the tabu tenure). During its tabu tenure, a tabu active reference point is not chosen as parent. This is a very simple form of recency-based memory within the tabu search methodology.

In general terms, the gap multiplier (1/3 in the example above), could be a random number r in the range (0,1). April et al (2003) express the use of linear combinations in OptQuest as

$$X_3 = x_1 - r\,(x_2 - x_1) \quad (4.8)$$

$$X_4 = x_1 + r\,(x_2 - x_1) \quad (4.9)$$

$$X_5 = x_2 + r\,(x_2 - x_1) \quad (4.10)$$

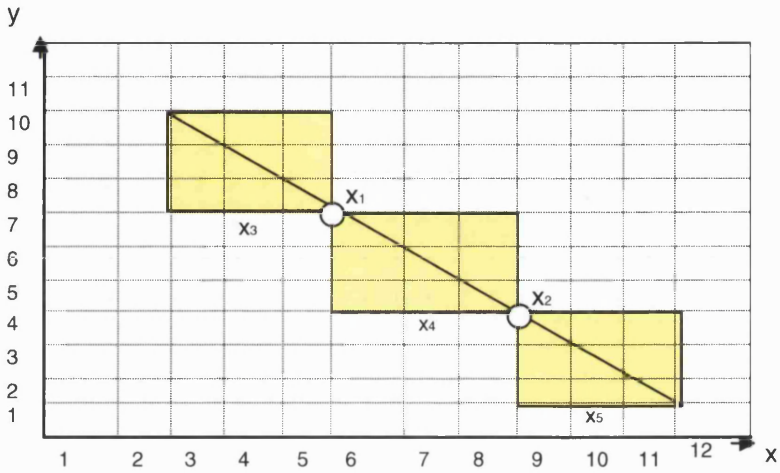


Figure 4.1.10 Linear combination of two solutions (April et al, 2003)

When a different random number is used for each variable in the solution, the combination mechanism creates new trial solutions by sampling from the rectangles shown in Figure 4.1.10 , which depict the combination of two solutions, x_1 and x_2, to generate x_3, x_4, and x_5 in a two-dimensional space.

4.1.12 Psychoclonal algorithm

Tiwari et al (2005), after examining the need of hierarchy theory, attempted to amalgamate pychoclonal algorithm with an optimisation algorithm to evolve a meta-heuristic method.

Psychoclonal algorithm is based on the concepts derived from human psychology and artificial immune systems theory. It emanates from the nature of motivation and the types of need that people experience during their lifespan. The basic idea of such a theory is that people have certain fundamental needs and that people are motivated to engage in behaviour that will lead to the satisfaction of these needs (Maier, 1965). According to the theory, people can enhance their levels only if the lowest level needs are satisfied. The five levels of needs, known as Maslow's pyramid, from lowest to highest are psychological needs, safety needs, social needs, growth needs and self-actualisation needs (He and Kusiak, 1997).

Tiwari et al (2005) developed the concept of clonal selection of artificial immune system (clonal AIS) algorithm. Similar to the way neural networks were developed from the nervous system, the immune system inspired the immergence of AIS as a computational intelligence paradigm. The AIS aims at solving a wide range of tasks related to complex computational and engineering problems such as pattern recognition, machine learning, and combinatorial optimisation.

Tiwari et al (2005) proposed a clonal AIS algorithm to optimise assembly sequences. Their proposal was based on the need and exploitation of the surrounding space, where a single member is optimised locally and the population yields a broader exploitation of the search space. As a whole the algorithm performs a greedy search.

As a novel approach as it may be, the psychoclonal approach is similar to other metaheuristic methods where near optimal solutions are obtained by systematic search from a population of possible alternatives.

4.1.13 Particle Swarm Optimisation (PSO)

This optimisation method has roots in two main component methodologies (Kennedy and Elberhart, 1995). It ties with artificial life in general, and to bird flocking, fish schooling, and swarming in particular. It also relates to evolutionary computation and has ties to genetic algorithm and evolutionary programming.

Birds and fish adjust their physical movements to avoid predators, seek food and mates, optimise environmental parameters such as temperature, etc. The algorithm began as a simulation of a simplified social milieu where agents were thought of as collision-proof birds and the original intent was to graphically simulate the graceful but unpredictable choreography of bird flock. The birds' movements depended on maintaining an optimal inter-individual distances.

As in the social- psychological metaphor, particle swarm algorithm is an adaptable algorithm. Each particle is influenced by the success of its topological neighbours (Kennedy and Spears, 1998). This external function provides a particle its neighbour of a given type. Onwubolu and Clerc (2004) distinguish neighbourhoods in three ways: a) *Social* neighbourhood that takes relationships into account, b) *physical* neighbourhood that takes distances into account, and c) *queens* that use an extra particle to summarise the neighbourhood instead of using the best neighbour/ neighbour's previous best of each particle. As an adaptive algorithm, the population individuals adapt by returning stochastically towards previous successful regions in the search space.

The particle swarm optimisation further improves its strategy to avoid being stuck in a local optima. It particularly uses the *no-hope/re-hope* processes. When the particle swarm optimisation process is unable to output the desired or expected value then the swarm is in a no-hope state. This can happen when the individual particles are not moving or when no effective movement is observed or when the algorithm is producing the same best value for a number of iterations. When the algorithm gets into a no-hope state, the only way out is to either accept the result or *re-hope* for a better result. Re-hope expands the swarm to check if there is still hope to reach a better solution. The re-hope strategies can be categorised as lazy descent method (LDM), energetic descent method (EDM), local iterative levelling (LIL), and adaptive re-hope method (ARM). More details can be found in Onwubolu and Clerc (2004), Zhang et al (2004), Cleric (2002), Cleric (1999), and He and Wei (1999).

As in other metaheuristic methods, particle swarm optimisation works on searching for near optimum solutions from a population of possible alternatives. Its application to the travelling salesman problem (TSP) has been documented but there is no indication to show that it can be equally effective in manufacturing simulation optimisation. Particle swarm optimisation mainly focuses on continuous optimisation problems.

4.1.14 Ant Colony Optimisation (ACO)

Ant colony optimisation (ACO) is a branch of a form of the artificial intelligence - swarm intelligence which studies the emergent collective intelligence of groups of simple agents (Liu et al, 2004; Dorigo and Caro, 1999). In a group of insects that live in colonies, such as ants and bees, an individual can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behaviour it shows. Most ants are blind, however, each ant while it is walking, deposits a chemical substance on the ground called pheromone. Pheromone encourages the following ants to stay close to previous moves. The pheromone evaporates over time to allow search exploration. Dorigo and Maniezzo (1996) illustrated the complex behaviour of ant colonies by looking at a set of ants building a path to some food. An obstacle with two ends was placed in their way such that one end of the obstacle was more distant than the other. In the beginning, equal numbers of ants spread around the two ends of the obstacle. Assuming that all ants have the same speed, the ants going around the shorter path return before the others (differential effect). With time, the amount of pheromone the ants deposit increases more rapidly on the shorter path thus more ants prefer this path (autocatalysis). The difference between the two paths (preferential path effect) is the result of the differential deposition of pheromone since the ants following the shorter path make more visits to the source. Because of the pheromone evaporation, pheromone on the longer path vanishes in time. Similar to the particle swarm optimisation, the ACO algorithm is based on this ant colony behaviour.

The functioning of an ACO algorithm can be summarised as follows: A set of computational concurrent asynchronous agents (a colony of ants) moves through states of the problem corresponding to partial solutions of the problem. They move by applying a stochastic local decision policy based on two parameters called *trails* and *attractiveness* (Maniezzo et al, 2004; Dorigo and Stutzle, 2002). By moving, each ant incrementally constructs a solution. When an ant completes a solution the ant evaluates the solution and modifies the trail value on the components used in its solution. This pheromone information will direct the search of the future ants.

Furthermore, an ACO algorithm includes two mechanisms: *trail evaporation* and, optionally, *daemon actions*. The trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralised actions which cannot be performed by single ants, such as the invocation of a local optimisation procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective.

ACO has been applied on problems that include quadratic assignment problem, job-shop scheduling, vehicle routing and timetabling (Roli, 2002). As in other metaheuristic optimisation methods, ACO works based on systematic search from a pool of alternative solutions. However, LCO's application to general manufacturing performance optimisation needs more thorough testing.

4.1.15 Combined procedures for Simulation Optimisation

One notable approach that uses a hybrid of optimisation methods – a *global guidance system*, a *selection of the best procedure*, and *local improvement*, was discussed by Pichitlamken and Nelson (2003). They start to explain their approach by assuming that the feasible region is convex and finite, that the region is non-empty, the performance measures are independent and identically distributed.

The global guidance system ensures the convergence of the search so that, given sufficient time, it reaches and selects one of the optimal solutions. Specifically the philosophy of Shi and Olafssons (2000) nested partition (NP) method is adopted for this purpose. NP works based on identifying a sequence of most-promising subregions from a population. When better solutions are found in the current most-promising region, then the region is partitioned for finer exploration. On the other hand when better solutions are found outside the current most-promising sub region, then the NP backtracks to a super region of it. The idea is to concentrate the computational effort where there appears to be good solutions but not be trapped locally.

The local improvement scheme is intended to provide an intensification component to the NP method. The idea is to improve performance where the population is large but good solutions are clustered, or where the population is large but the response surface is smooth. The local improvement helps the search to explore more intensively near good solutions. A hill climbing (HC) algorithm was used for the local improvement. The current solution on hand is compared with some (or all) of its neighbouring solutions and the winner becomes the next solution. This neighbourhood selection of the best is repeated until some stopping criterion is satisfied.

Each NP iteration and each HC step requires selecting the best solution from among a number of candidates i.e., the sampled solutions for NP and the neighbouring solutions of the current best for HC). Sequential Selection with Memory (SSM) is used to provide a highly efficient method for selecting the best optimum performance while controlling the chance of an incorrect selection. SSM is fully sequential with elimination, which means that it takes simulation outputs one at a time from the solutions under consideration, and eliminates solutions as soon as they are shown to be inferior. SSM is particularly designed to investigate revisit solutions because it exploits whatever data already obtained.

Morito et al (1999) also attempted to combine mathematical optimisation methods with metaheuristics to achieve optimum model conditions. The simulation based constraint generation, as applied to logistics may not be a true match to manufacturing systems though.

4.2 Data Consistency Analysis

4.2.1 Introduction to control charts in model consistency.

In a manufacturing performance analysis process, it is desirable that the data and the process of a given system are predictable. In a performance improvement sense, the system's variability has to be minimised to an acceptable level before effective performance improvement processes can commence. According to Deming (2000), a process with no indication of special (systematic) causes of variation is said to be in statistical control. Although it is a random process, its behaviour in the near future is predictable. A system that is in statistical control has a definable identity and a definable capability. In the state of statistical control, all special causes so far have been removed. The remaining variation is left to chance – i.e., to common causes. The next step is to improve the process. Improvement of the process can be carried out effectively, once statistical control is achieved and maintained i.e., once the output variability has been minimised to an acceptable level.

Since most simulation models use random variables as input, the simulation output data are themselves random and can be properly assessed by conducting simulation runs. In most operational systems a lower level of variability is preferred since it is easier to match resources to the levels demanded (Robinson, 2004D). Indeed, a worse value with low variability may be selected in preference to a better value with high variability. Generally, if the variability of the output is high, minimising the variability is recommended prior to considering investment on resources to improve performance.

Minimising variability

The variability of an output is dependent upon the distribution functions of influencing input parameters. One way of minimising variability of outputs is by conducting sensitivity analysis whereby action is taken on the input parameters that have higher sensitivity towards the variability of the output (Robinson 2004E). The concept is shown in figure 4.2.1. The input (*I*) is varied and the effect on the response is measured. If there is significant shift in the response (the gradient is steep), then the response is sensitive to the change in the input. If there is little change (the gradient is shallow), then the response is insensitive to the change.

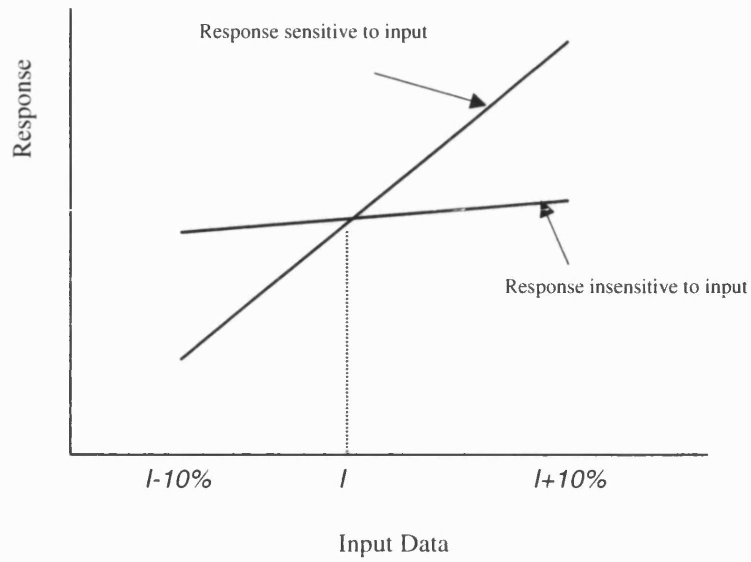


Figure 4.2.1 Sensitivity Analysis (Robinson, 2004E)

In view of simulation modelling, the main approach to performing sensitivity analysis is to vary the model inputs, run the simulation and record the changes in the response. This could be a time consuming process especially if there are many model inputs. For this reason, sensitivity analysis is generally restricted to a few key inputs, which may be identified as those about which there is the greatest uncertainty and which are believed to have the greatest impact on the response (Robinson, 2004E; Law and Kelton 2000E; Balci, 1994).

Once the variability of the simulation outputs has been minimised to an acceptable level, the process of performance optimisation can proceed more effectively.

Control Charts

In most manufacturing and engineering applications, it has been accepted that the output from processes can be modelled as a normal (Gaussian) distribution curve and predictions based on that model can be made with a great deal of confidence (Walker, 2005).

One of the most widely used tools for assessing invariability of data is the *control chart*, which is a graphical display representing a series of samples taken from a process and where evaluation is made whether the process continues in a state of statistical control or not. A typical control chart will show random variation of the data around the process mean. If the process data points are not at random around the mean then this may indicate a problem with the process that needs attention. Among the most commonly used types of control charts are the *mean chart*, the *range chart* and the *p-chart*. A *cusum chart* is also a type of a control chart that has its own merits over the others. A brief account of these control charts is given in the next section.

Other control charts for variables include run charts, multi-vari charts, moving average charts, moving range charts, median and mid-range charts (Montgomery 2001; Miller et al 1990; Owen 1989; Oakland 1996). Attribute based process control charts include np-charts, c-charts, and u-charts.

4.2.2 Mean and Range Charts

A typical *mean chart* would have graphs of the mean, action limits, warning limits, and the raw process data. A mean chart, as recommended by ISO 8258 (1991) works on a recommended minimum sample size of 25 to safely assume that the actual distribution is normal. A typical mean chart is as described in Figure 4.2.2.

LAL and UAL are lower and upper action limits respectively, and
LWL and UWL are lower and upper warning limits respectively.

The limits are calculated from:

$$\text{Lower limit} = \mu - z_{a/2} \cdot \sigma/\sqrt{n} \tag{4.11}$$

$$\text{Upper limit} = \mu + z_{a/2} \cdot \sigma/\sqrt{n} \tag{4.12}$$

Where μ is the mean, σ is the standard deviation, and n is the sample size.

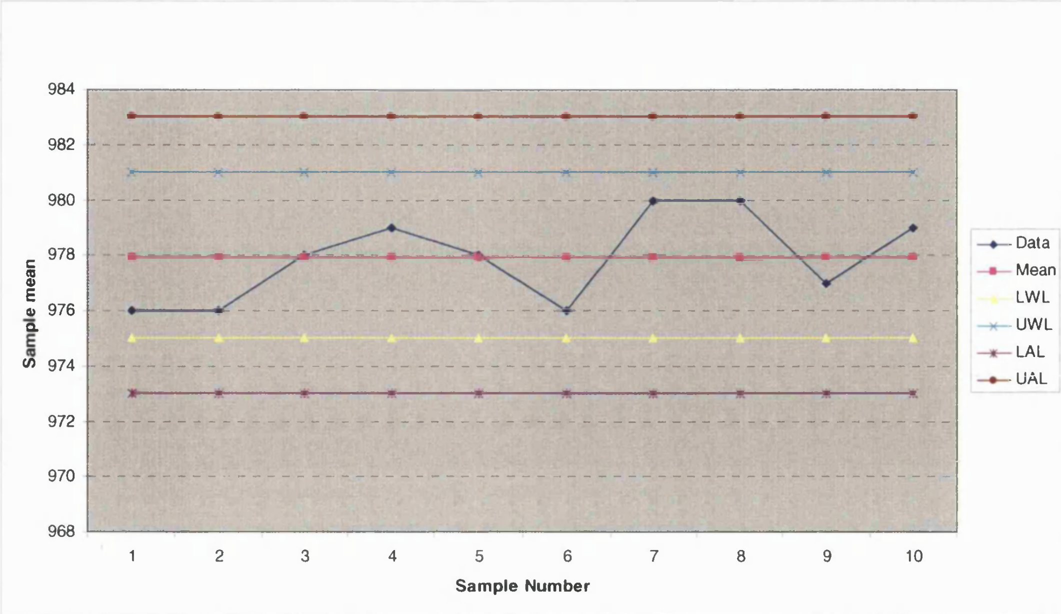


Figure 4.2.2 Mean Chart

BS 2564 uses $z_{a/2}$ as 3.09 (for actions limits) and 1.96 (for warning limits) to give the probabilities of 0.999 and 0.975 respectively. ISO standards (ISO8258, 1991) locate the limits as:

Action limit lines at $\mu \pm 3\sigma/\sqrt{n}$

Warning limit lines at $\mu \pm 2\sigma/\sqrt{n}$

In broad terms, for the system data to be considered under control, none of the data should go beyond the two action limits, and no two consecutive points in ten should lie beyond the warning limits. A detection of seven runs above or below the mean value and/or trends of seven rising or falling are also treated as the system going out of control.

A range chart is similar to the mean chart except that it uses ranges from replications of each sample. It is often used to supplement the mean chart in decision making. The range chart only picks up long term trends and is rarely used in industry. The Upper action limit and the upper warning limit lines are often of importance to the range chart, and are calculated using a simple approximation for industry as

Upper action limit line at $D`_{.001} R_{av}$

Upper warning limit line at $D`_{.025} R_{av}$

Where $D`_{.001}$ and $D`_{.025}$ are constants read from a table, and R_{av} is the average of the ranges. For further reading please see Oakland (1996).

4.2.3 The p-chart

One of the commonly used control charts for attribute data is the p-chart. Attribute data are expressed in a form of *conforming* or *non-confirming*, *acceptable or defective, go or no go* etc.. The probability of finding x defectives in a sample of size n when the proportion present is p, is generally represented by a binomial distribution as

$$P(x) = \binom{n}{x} p^x (1-p)^{(n-x)} \qquad (4.13)$$

The first step in the design of a p-chart is the calculation of the average proportion defective ($\bar{p}$):

$$\bar{p} = \sum_{i=1}^{k} x_i / \sum_{i=1}^{k} n_I \qquad (4.14)$$

Where k is the number of samples, and

$\sum_{i=1}^{k} x_i$ is the total number of defective items

$\sum_{i=1}^{k} n_i$ is the total number of items inspected.

The control charts for a p-chart are calculated as

Warning limits = $\bar{p} \pm 2\sigma$ (with BS5703, $\bar{p} \pm 1.96\sigma$) (4.15)

Action limits = $\bar{p} \pm 3\sigma$ (with BS5703, $\bar{p} \pm 3.09\sigma$) (4.16)

A typical p-chart is shown in Figure 4.2.4.1.

4.2.4 Cumulative-sum (Cusum) Control Charts

The mean and range control charts, also known as Shewhart control charts after the man who first described them in 1920 (Oakland, 1996), are mainly concerned about the plots and basic investigative rules for deciding whether the points lie within the control limits or not. Process control by Shewhart charts considers each point as it is plotted.

Unlike Shewhart charts, the *cusum* method employs a technique which uses all of the information available. The short- and long-term changes and trends of a process could be detected with a cusum chart because the cusum chart takes account of the past data.

The cusum method which was developed in the United Kingdom (BS 2564, 1955, BS5703 part 3, 2003) is one of the most powerful tools available for the detection of trends and slight changes in data. The cusum chart is recognised as the best statistical process control chart for detecting small changes in performance that would not be detected by the Shewhart chart (Owen, 1989). Cusum charts are also most advantageous when dealing with a population considered to be homogeneous and a sample of one could be acceptably taken (Oakland 1996, Walker 2005).

The cusum chart can be best understood with an example. Table 4.2.4 shows the number of minor machine breakdowns per month. Looking at the figures alone would not give the reader any clear picture of the level of machine failures of the company. Figure 4.2.4.1 is a Shewhart chart (p-chart) on which the results have been plotted.

Looking at the chart, the average number of failures is 3.1 and the process is obviously in statistical control since none of the sample points lie outside the action line and only one is in the warning limit. It is difficult to see from this chart any significant change, but careful examination will reveal that the level of failures is higher between months 2 and 17 than that between months 18 and 40. However, only individual data points can be seen on the chart.

Table 4.2.4 Number of minor machine failures per month in a large company

Month	No of failures	Month	No of failures	Month	No of failures	Month	No of failures
1	1	11	3	21	2	31	1
2	4	12	4	22	1	32	4
3	3	13	2	23	2	33	1
4	5	14	3	24	3	34	3
5	4	15	7	25	1	35	1
6	3	16	3	26	2	36	5
7	6	17	5	27	6	37	5
8	3	18	1	28	0	38	2
9	2	19	3	29	5	39	3
10	5	20	3	30	2	40	4

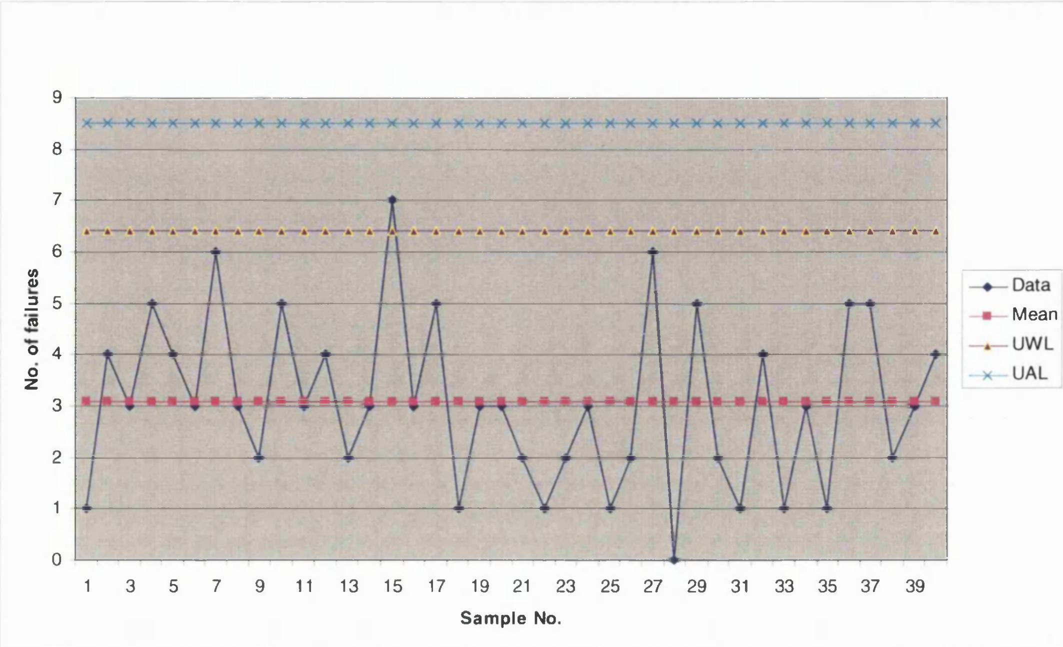


Figure 4.2.4.1 Shewhart chart (p-chart)

In Figure 4.2.4.2, the same data are plotted as cumulative sums on a cusum chart. The average number of defectives, 3.1 (also denoted by T the mean target value), has been subtracted from each sample result and the residues cumulated to give the cumulative sum. That is, a column of x-T values of all the data was created and then another column with cumulatives of x-T was created to give cusum values.

In the cusum chart, the difference in failure levels is shown dramatically. It is clear, for example that from the beginning to month 17, the level of minor failures is on the average higher than 3.1 since the cusum plot has a positive slope. Between months 18 and 35 the average failure level has fallen and the cusum slope is negative. Manual calculations for the averages of months 1-17 and 18-35 give 3.7 and 2.3 respectively confirming that the signal from the cusum chart was valid. This information calls for investigating the cause of the persistent increase in failure level in the months 1 to 17 which using the Shewhart chart could have been overlooked.

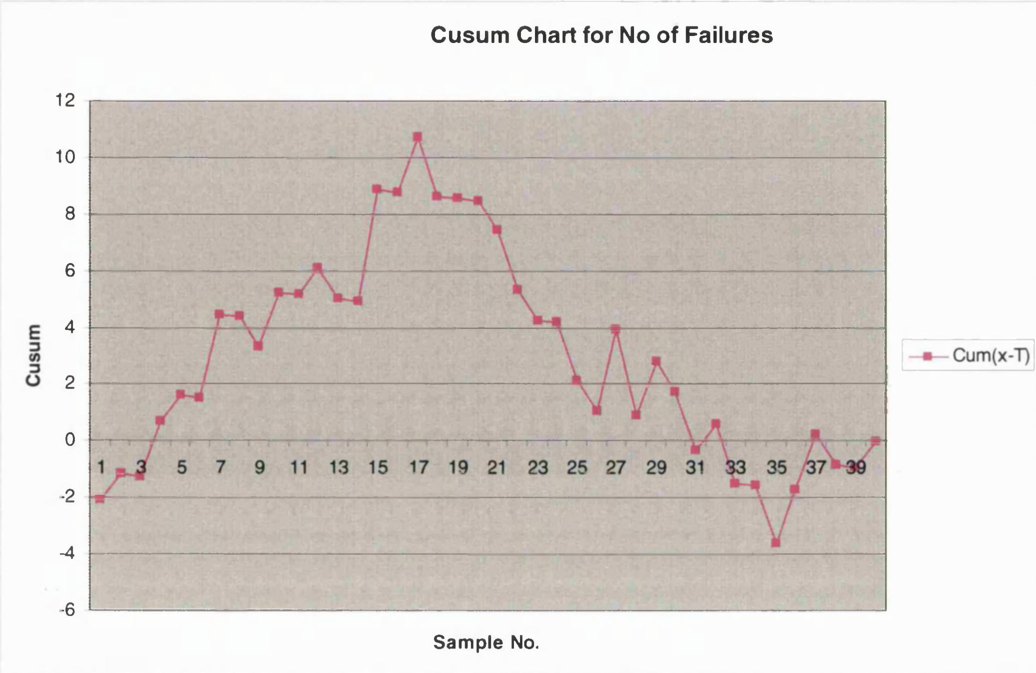


Figure 4.2.4.2 Cusum Chart for Number of Failures

Oakland (1996) explains that cusum is about the slope:

- if the cusum slope is upward shows the observations are above the target
- if the cusum slope is downward shows the observations are below the target
- if the cusum slope is horizontal shows the observations are on target
- the absolute value of the cusum score has no meaning.

Cusum decision procedures

The extreme sensitivity to change of cusum charts needs to be controlled if unnecessary adjustments to the process and/or stoppages are to be avoided. The largely subjective approaches, as seen earlier, are not very satisfactory. It is desirable to use objective decision rules similar to the control limits on the Shewhart charts, to indicate when significant changes have occurred. The two main methods with practical application in industry are the *V-mask* and *decision intervals (BS5703, 2003; Coleman, 1996; Oakland,1986)* . Since the two methods are theoretically equivalent, despite different mechanics, only the V-mask will be looked into here.

V-masks

The most common method of checking the slope to determine if the process is going out of control is the V-mask (see Figure 4.2.4.3). The V-mask is usually drawn on a transparent overlay and then superimposed on the cusum plot. If all the data lie within the arms of the mask then the process is under control. If any of the data is obscured by the mask i.e., if the cusum graph crosses or touches either of the upper or lower mask lines, then the process is out of control.

The V-mask C1, recommended for general use (Coleman 1996, BS5703 2003) is 10 samples long and $\pm$ 10s high at the wide (left) end and $\pm$5s high at the narrow end, as shown on Figure 4.2.4.3. This is based on

Action limits at $\quad T \pm 3s/\sqrt{n}$

Warning limits at $\quad T \pm 2s/\sqrt{n}$

Where s is the standard deviation of the samples taken.

Another standard mask suggested by BS5703: Part 3 is C2 which is based on

Action limits at $\quad T \pm 2.65\ s/\sqrt{n}$

Warning limits at $\quad T \pm 1.65\ s/\sqrt{n}$

C2 is based on the probability of an action limit of 1 in 250 and a warning limit of 1 in 20. The C2 mask design is 8.5s, 3.5s and 10 readings.

As an improvement to the standard masks, semi-parabolic masks can be used. They are calculated based on standard errors but with improved detection of large shifts with little increase in the risk of a false alarm. The masks are plotted using standard error values, dependent on distance from the focus point, which are obtained from a table (see Appendix D). These standard errors are multipliers of s at each point. An example of an experimental simulation output data with varying random stream values which proved to be in statistical control using the C2 mask is shown in Figure 4.2.4.4.
A good application of cusum charts in an engineering application can be seen in Walker (2002).

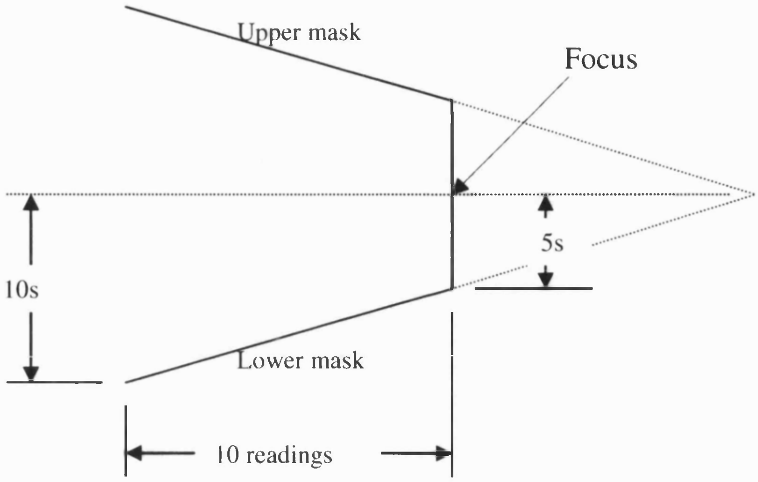


Figure 4.2.4.3. Typical V-mask (Walker, 2005)

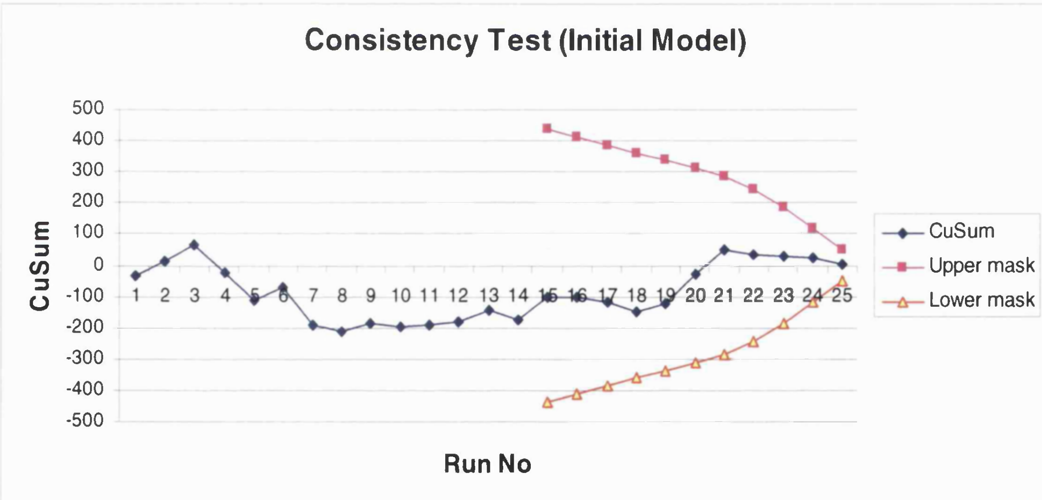


Figure 4.2.4.4. Cusum check for a simulation output variability

4.3 Review of Existing Optimisation Methods

Going down the list of optimisation methods, there is a transition from complete confidence in optimality in an unrealisable context to workable solutions that apply in practical settings. It can be said that both research and practice have adapted approaches from deterministic optimisation:

- Gradient search :- move locally in the most promising direction according to gradient
- Random search:- move randomly to a new point, no information is used in the search
- Response surface methodology:- function approximation by regression.
- Simulated annealing:- sometimes move in a locally worse directions to avoid being trapped in local extrema
- Genetic algorithms and scatter search:- population based, generates new members using operations on attributes of current members
- Tabu search:- uses memory (search history) to avoid tabu moves
- Neural networks:- non-linear function approximation
- Mathematical programming:- powerful arsenal with rigorously tested software.

The most commercially applied optimisation methods are based on metaheuristics, predominantly evolutionary algorithms, that perform some sort of search for optimal outputs for a set of input parameters (see table 4.3). Yet, the procedure usually needs to run a large number of alternative models before deciding on the best solution, and it often requires a substantial amount of computer power and time. Referring to the use of AutoStat as an optimiser, Birton (2000) said, "*Optimisation analyses take a large number of runs. You can use AutoStat to make runs on multiple machines on your network... You can take advantage of other machines to make runs overnight or on weekends*".

Fu (2002) reviewed five commercially applied simulation optimisation systems, with the goal of using their routines to seek improved settings of user-selected system parameters with respect to the performance measure(s) of interest. He concluded that contrary to the use of mathematical programming software packages, the user has no way of knowing if an optimum has actually been reached. He added, the biggest problem with currently implemented optimisation methods is that though they may be intelligent in performing the search procedures, they are somewhat oblivious to the stochastic nature of the underlying system. Thus, they completely lack any sense of how to efficiently optimise a simulation. Fu (2000) exclaims, it is baffling that RSM, the well established regression method which is quite general and easy to implement has not yet been incorporated into any of the commercial packages.

Table 4.3 Commercial Optimisation Packages

Optimisation Package (Simulation Platform)	**Vendor**	**Primary Search Procedure**
AutoStat (AutoMod)	AutoSimulations Inc. www.autosim.com	Evolutionary, genetic algorithms
OptQuest (Arena, Crystal Ball, Microsaint, Taylor Enterprise Dynamics, QUEST, SIMPROCESS)	Optimisation Technologies Inc www.opttek.com	Scatter search and tabu search, neural network
OPTIMIZ (Simula8)	Visual Thinking International Ltd www.simul8.com	Metamodelling with Neural networks (Now discontinued)
SimRunner (Promodel)	PROMODEL Corp. www.promodel.com	Evolutionary, genetic algorithms
Optimiser (WITNESS)	Lanner Group www.lanner.com	Simulated annealing , tabu search

Much of the existing research has concentrated on relatively narrow areas or toy (non-realistic) problems, the single-server queue being the most obvious example. Most of the works lack the jump to the next step in practice. Additionally, most of the optimisation methods discussed above have not been properly applied in industries due to their requirement of a high level of sophistication on the part of the user. April et al (2003), while describing the amount of attention given to simulation optimisation from the research community, expressed the view that the analytical techniques (such as response surface methodology) require considerable technical depth such that they have not been favoured by industrial users.

Two aspects that may favour some of the available optimisation software are (1) The generality of the optimisation packages could enable them to handle a wide range of problems that the user is likely to encounter, and (2) with graphical user interfaces and pull down menus, the mathematical requirement of users has seen a marked reduction. While this has the clear benefit of allowing the power of simulation to reach a much wider audience of users, it also means that any complications associated with optimisation must be shielded from the interface. In most cases, by treating the simulation model in the way that metaheuristic approaches are generally applied, there is an immense waste of simulation replications used to obtain precise estimates at variable settings whose poor relative performance becomes apparent with just a few replications.

The use of the performance optimisation/enhancement techniques described above has been well documented. The commercial methods such as Optimiser and OptQuest, as applied to manufacturing simulations, have proven to be good search engines and can in certain cases produce satisfactory results. But their use mainly depends on systematic search from a pool of possible models rather than employing the practical expertise-oriented methods that manufacturing engineers would use (such as Theory of Constraints) which in most instances leads to improved settings with a high level of confidence. Boesel et al (2001) and Fu (2000) expressed their concern about problems that may arise from the lack of understanding of managers on how integrated simulation-optimization systems work.

An alternative method, to be discussed in the following chapters, is to employ an expert mechanism that will make use of proven operations management performance enhancing techniques. This approach is aimed at helping users from manufacturing personnel (such as operations managers and manufacturing engineers) to understand the optimisation process and have more confidence in the solutions obtained. Additionally, unlike search strategies, the process will not waste time dealing with sifting through a population of simulated results which usually requires significant time and computer power.

4.4 Review of data consistency checks in simulation

The author has not come across any published work that looks into appropriate data consistency checks on simulation model results. As the most common manufacturing models are stochastic, it is understood that the simulation output values will also be variable. For this reason it is necessary to check that the output data variability is reduced to an acceptable level before optimisation can commence.

Among the ways to check model data consistency is to use the cusum chart method. Cusum's superior ability to detect small output changes and its clear graphical representation make it among the best control techniques for this purpose.

CHAPTER 5
AIMS, OBJECTIVES AND METHODOLOGY

This chapter re-states the research aims, and following the literature review and discoveries in the previous chapters, describes the research objectives, and discusses the research methodology used to achieve the research objectives.

5.1 Aims

The main aims of the research are

a) To conceive, develop and evaluate a system, integrated to a manufacturing simulator, that can assist in performance analysis and optimisation of a manufacturing system.

b) To derive a method to check simulation model data consistency and to evaluate solutions.

The main part of this research project focuses on the mechanism that supports a manufacturing simulator in interpreting simulated results, assessing performance and taking action to enhance performance. The powerful features of modern simulation packages are generally concentrated on the front-end activities of building models easily and on getting simulation results quickly. As a result, large reports and large amounts of raw data are generated but do not help the user see how these reports inform decisions about which appropriate action to take in a consistent and logical way. Additionally, limited alternative simulation models could be dealt with in a traditional way, but as alternatives increase, conducting a large number of simulation runs becomes time consuming and costly.

Some commercial simulation packages now include some types of integrated optimisation routines that generally work on the systematic search of optimum values from a pool of simulated results but do not make use of proven manufacturing engineering methods. Thus they are not usually user friendly to industrialists, require powerful computers and take a long time.

The proposed work is expected to be based on manufacturing engineering/operations management expertise embedded in the expert mechanism which would be integrated into a manufacturing simulator. This method generally takes an action which is clearly understood by industrialists, and is more likely to enhance performance thus reducing the number of simulation runs and the time required.

The second aim deals with checking data consistency of the simulation output data.
In a performance improvement sense, a system's variability has to be minimised to an acceptable level before effective performance improvement process can proceed. That is, the improvement process requires that statistical control is achieved and maintained. It is proposed that a method similar to control charts could be used to make sure that the simulation output data is statistically consistent.

5.2 Objectives

To realise the aims the following objectives were identified.

- To conduct a literature survey on manufacturing simulation as a manufacturing performance analysis tool and as applied for performance enhancement; to verify originality of project;
- To identify a suitable manufacturing simulation system and examine the relevant elements of the simulation system that could be used in manufacturing analysis.
- To build representative models and examine the influencing factors;
- To identify a suitable method of automatically controlling the parameters that affect the performance of manufacturing systems;
- To develop an expert mechanism that assists in automatic interpretation of manufacturing simulation results and that effects changes to input data for optimising purposes prior to subsequent simulation runs;
- To integrate the expert mechanism with the simulator;
- To conduct experiments to verify and validate the system;

5.3 Methodology

This sub-chapter describes some of the available research methods and why particular methods have been considered applicable to this research project.

5.3.1. Overview of Research Methods

Research is a scholarly or scientific investigation or enquiry that requires thorough study so as to present in a detailed and accurate manner (University of Bath, Mechanical Engineering course on Research Methods - ME50173, 2006). Carrying out of research has two elements:

- empirical knowledge: acquiring data, observations, facts, cases, etc.
- theoretical knowledge: laws, principles, models, concepts, etc.

Generally, a research process follows either a *deductive* or *inductive* approach. The deductive approach first finds a theory (or proposal) and is then tested with data. This is more appropriate to most engineering types of research. The inductive approach gathers data and then thinks of a theory. The inductive approach is more suited to social sciences and humanities research.

There are a number of Research classifications in the literature. Clarke (1972) and later, Howard & Peters (1990) classify forms of research as pure, basic objective, evaluative, applied and action. Alternatively, Philips and Pugh (1987) argued that the classification of pure and applied research is too simplistic and preferred to classify research as exploratory, testing out and problem solving research. However, within each of these classifications certain methodological problems have to be considered and resolved. These concern how to aggregate different clusters of independent data, the relative importance of analysing data gathered at different levels, and the wider issue of sampling frames for data collection (Bryman, 1989).

Trafford (2001) writes in appreciation of Kuhnian notion of paradigms which explain and produce significant shifts in understanding. Kuhn (1962A) suggested that scientific paradigms are examples of actual scientific practice, examples which include law, theory, application and instrumentation together ... to provide models from which spring particular coherent traditions of scientific research. According to Kuhn, paradigms are also the source of the methods, problem field, and standards of solution accepted by any mature scientific field at any given time (Kuhn, 1962B).

Burrell and Morgan (1979) present four assumptions about the nature of research: ontology, epistemology, human nature, methodological. This approach is more inclined towards research in social studies and includes the idea that hypotheses could be expressed which try to capture theoretical explanations of practice – by researchers who have incorporated these assumptions about how their research has been designed.

Rose (1982) produced a model which is also represented by Traford (2001) that shows how the key components of research are systematically related to one another by linking theory and evidence. He developed an ABCDE model as shown in Figure 5.3.1

A. ***Theory***: an explanatory statement about the phenomena
B. ***Theoretical propositions***: specific propositions to be investigated in the study
C. ***Operationalisation***: decisions made on how to carry out empirical work; technique of data collection; sampling; concepts and indicators, variables; units
D. ***Field work***: collecting data, practical problems of implementing stage C decisions
E. ***Results***: data analysis leads to findings; interpretation feeds back to C,B,A.

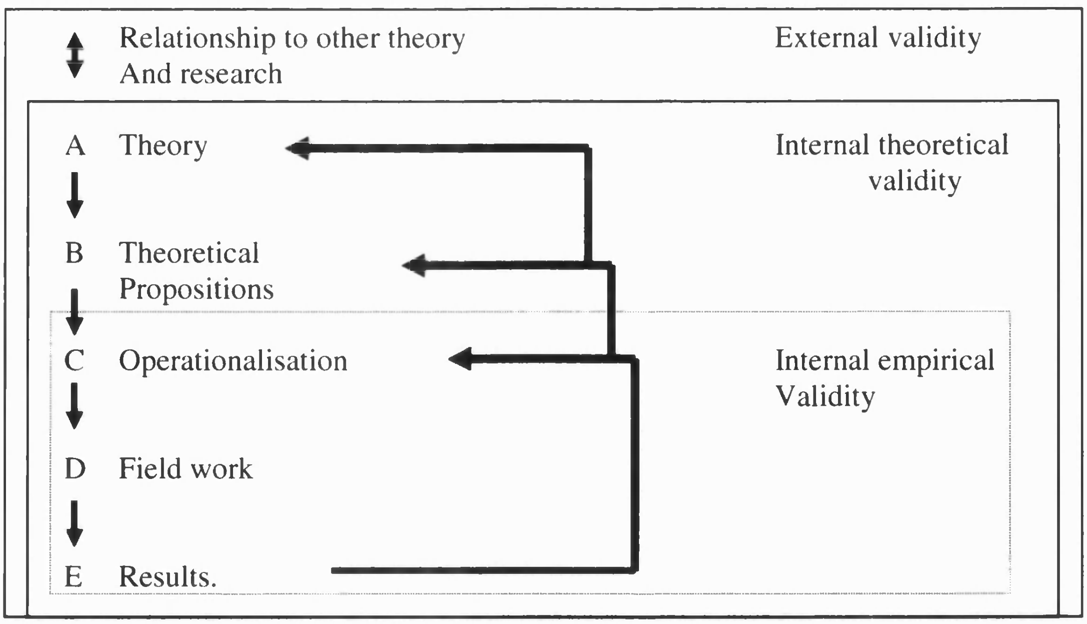


Figure 5.3.1 Rose's ABCDE model and distinction between three kinds of validity in Research (Trafford, 2001)

The model illustrates how researchers have justified progressing through each stage from theory to results. Rose developed the route further by indicating that by tracing back through the E-C-B-A route the validity of the research process can be evaluated. The significance of the model is that the C point in each model is central both to the developmental process as well as to the evaluative processes. The research assumptions from A to C relate to conceptual issues, whilst those from C to E relate to operational issues.

An approach described by Walton and Gaffney (1991) specify a research cycle that comprises the following stages:

1) Identification of a study topic
2) Operationalisation of a hypothesis
3) Selection of an observation sample
4) Selection of a research method, gathering of data and generation of findings
5) Derivation and dissemination of the implications for theory and practice.

5.3.2 Research methods in Engineering

Although research methods in science and in engineering have plenty in common, according to the University of Bath, Mechanical Engineering course in Research methods - ME50173 (2006), they have some conceptual differences that include

- engineering incorporates science but also rules of thumb
- engineering is "know how" not "knowing that"
- engineering seeks safety but science seeks truth
- engineering tries to avoid being refuted yet science tries to refute.

Blockley & Henderson (University of Bath, Mechanical Engineering course in Research methods - ME50173, 2006) describe engineering processes as having the following steps

- encounter a problem
- propose a solution
- assess the consequences
- decide how to embody the solution

- embody it
- test it
- learn how dependable it was.

As an extension of Blockley & Henderson's description of engineering processes, for a specific piece of engineering research, the slides of ME50173 (2006) show the following steps

- fix the basic area of the work
- find out what is already known (review of previous work)
- identify the problem or gap exactly (problem definition and generation of aims)
- develop a precise objective
- perhaps propose and build a trial artefact
- collect data on its performance
- analyse the data
- draw conclusions
- disseminate findings.

5.3.3 Methods for this Research Project

This research project, as described in sections 5.1 and 5.2, deals with an applied manufacturing problem and thus falls into the engineering research category. The initiation, research area and broad description of the research title have been explained in Chapter one, section 1.3.

Literature Review

Literature survey includes primary (such as archival journals, theses and dissertations), secondary (review journals, monographs and textbooks) and tertiary (indices, catalogues, encyclopaedias, bibliographies) literature. The literature review conducted in this research included current review from the academic and industrial points of view.

Literature on manufacturing processes, manufacturing performance analysis, simulation and modelling, simulation in manufacturing, manufacturing performance optimisation, and simulation optimisation were thoroughly surveyed and those relevant to the research reviewed (chapters 2, 3 and 4). The literature review has been instrumental in shaping the research aims and objectives.

Problem definition and research aims

The problem posed has been broadly defined in the Introduction section and summarised at the start of this chapter (section 5.1). Following an extensive literature review, the gaps in the manufacturing simulation body of knowledge were clearly identified and the research aims generated. The research aims can be summarised as a) a technique can be created that could assist production engineers and managers in interpreting results, and in assessing and optimising performance when using manufacturing simulation. The technique could include the use of proven performance enhancing methods that are not alien to manufacturing engineers and managers. b) There is a need for a method to check data consistency and evaluation of a manufacturing simulation model.

Developing precise objectives

The objectives of the research have been described in section 5.2 of this chapter.

Proposing and building a trial model

A proposal for finding solutions to the research problems included the use of a rear-end mechanism to interpret simulation output, to assess performance and to enhance performance. The proposed model is shown in more detail in chapter 6.

For preliminary trial, a suitable simulation package was identified, and then a verified case study model was used. The relevant input data and relevant reports were investigated, and the performance influencing factors identified. Sufficient number of runs were conducted and, based on the output data, consistency of the output data was checked manually. This was followed by analysing results and manual application of performance enhancing methods. The experimentation continued until the results were satisfactory.

Once the reliable experimental results had been obtained, a suitable programming software for creating the expert mechanism was identified. The data consistency checking rules and the performance enhancing rules were then written and the expert mechanism (program) integrated with the simulator. Further verification and validation of the system was then done through case studies.

Data collection and analysis

This research focuses on applied manufacturing simulation solutions to industry. As the basis to examine the industrial application of the methods and findings, case studies were the core sources of data. A case study with a verified model from the simulation package vendor, Lanner Group, was the first set of data used for experimentation. This was useful to verify and partially validate the expert mechanism. Another industrial case study was used to build a second step model with more features. The third case study from a world class company, SELEX AS & S, tested the robustness and validity of the findings. A comparison in performance of the expert mechanism against one of the commercially popular optimisation systems was done to validate the expert mechanism.

Conclusions and dissemination of findings

Analysis and findings of experimental data from the case studies helped to conclude that the research substantiated the hypotheses. The findings are disseminated as a thesis, publications in conferences and journals, and a company report.

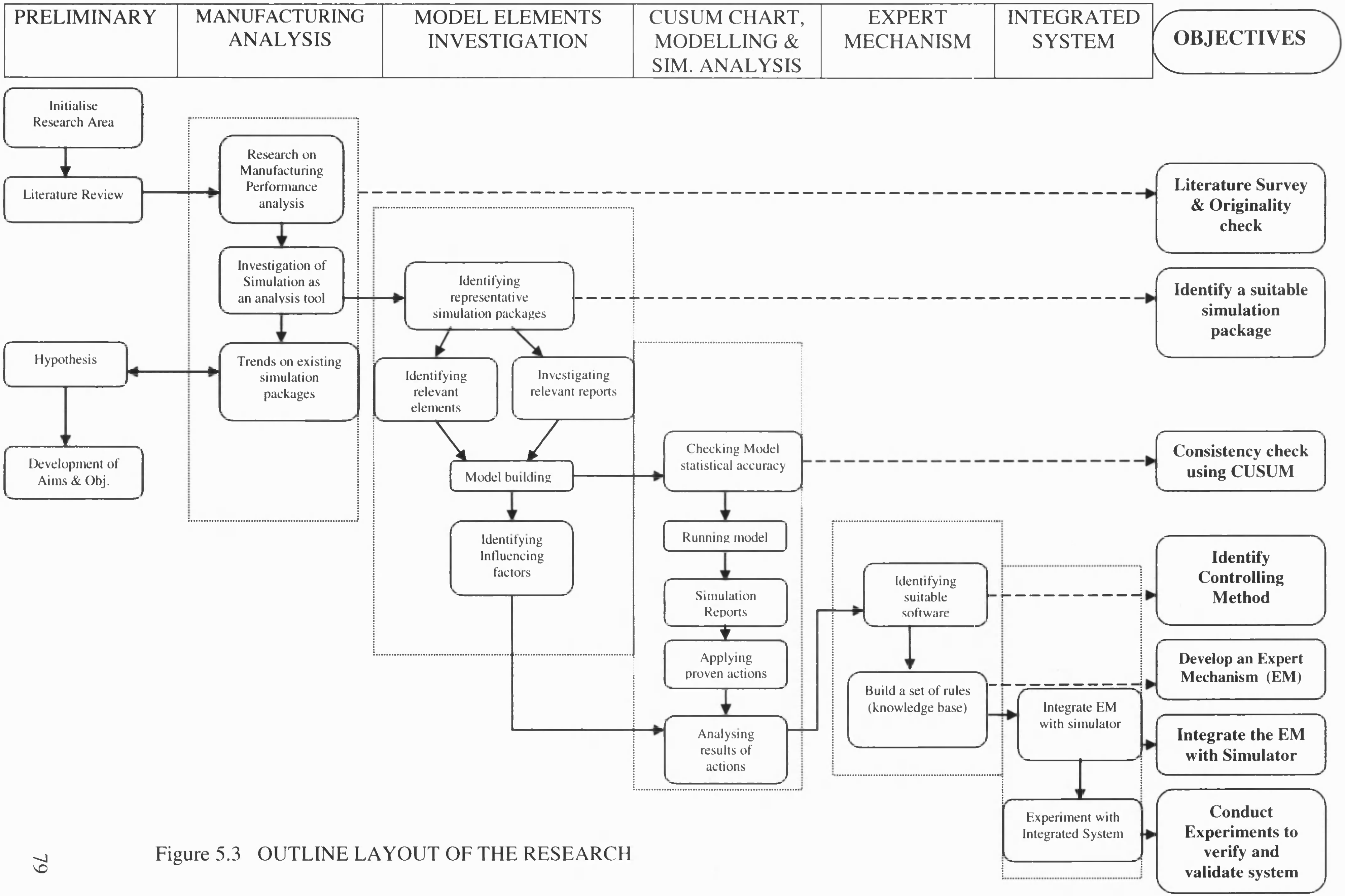


Figure 5.3 OUTLINE LAYOUT OF THE RESEARCH

CHAPTER 6
PROPOSED SYSTEM

This chapter, consistent with the research aims and objectives, discusses the technical structure of the proposed system. It describes the simulator, the data consistency checking element and the optimisation element. It also depicts how these elements are integrated.

6.1 Summary

The proposed system has a manufacturing *simulator*, a data *consistency check element* and an *optimisation* element. The purpose of the manufacturing simulator is to build a representative model of a manufacturing system which is run to generate realistic outputs. Then the data consistency checking element would take experimental results from the simulator and check that the system output(s) is under statistical control i.e., it would check that the variability of the simulation output is within an acceptable level. Once the output data consistency has been established, then the optimisation element would take over and conduct performance enhancement procedures. A summary of the proposed system is shown in Figure 6.1

6.2 Simulator

6.2.1 Objective of Simulator

The main objective of the simulator is to represent (imitate) the manufacturing system in question whereby with appropriate inputs, it would generate outputs that are necessary to assess the system performance for a given scenario. The most common elements of a manufacturing simulator include entities (parts), resources (machines, operators, material handling equipment), queues (buffers). Entities have attributes that distinguish them from each other. An entity moves through the system and requests the use of resources. If a requested resource is not available, then the entity joins a queue. The priority of entities in a particular queue depends on the priority dispatching rule

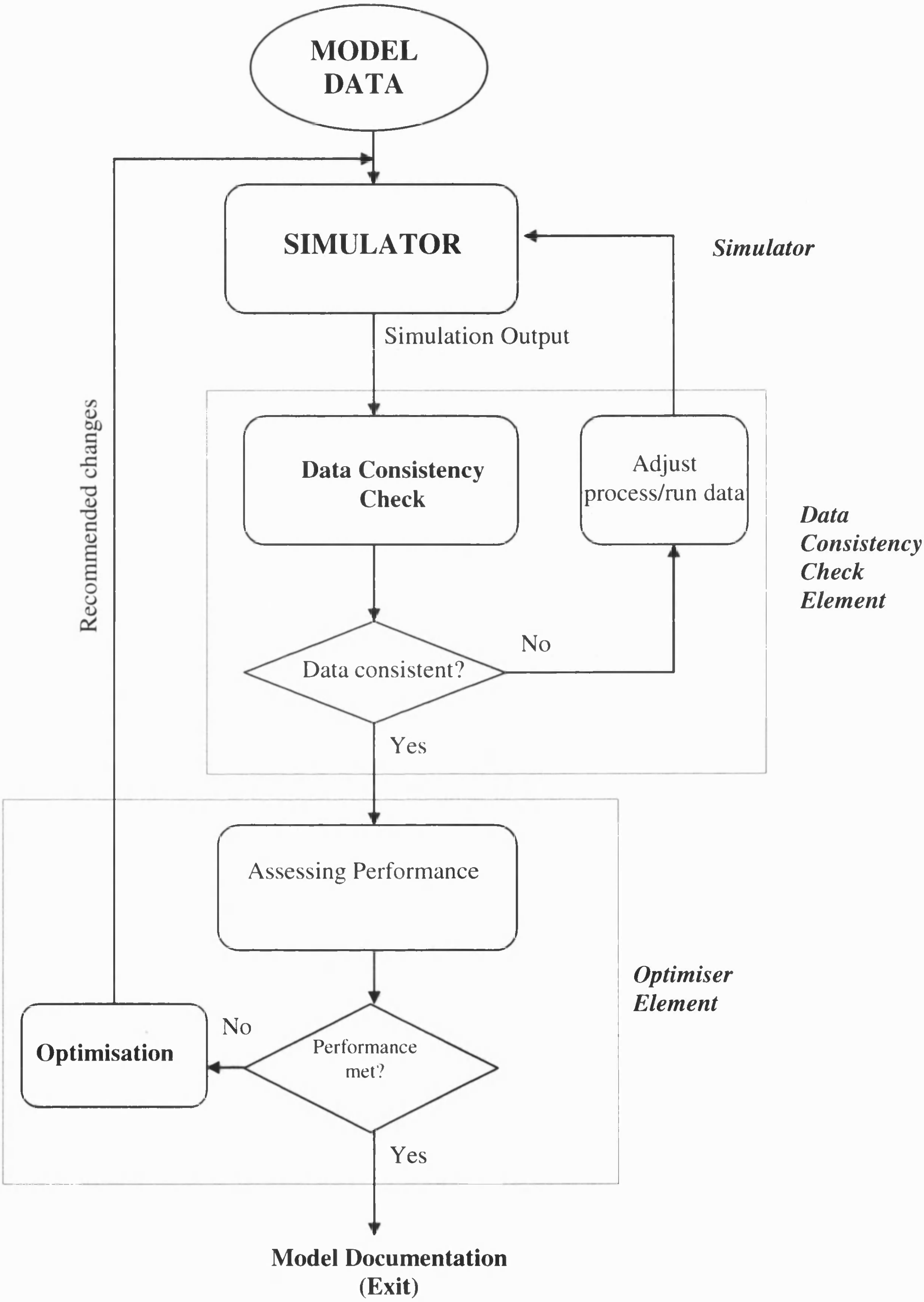


Figure 6.1 Summary of Proposed System

used (FIFO, LIFO, etc). The process time and other effects for an entity when it seizes a resourse is detailed in the operation element.

Inputs to a manufacturing simulator are those that can be altered to effect changes in performance of the system. The common input data to a manufacturing simulator include

- Interarrival time of parts
- Batch size of parts
- Quantity of resources (machines, operators)
- Buffer sizes
- Shift schedule
- Process times at each resource
- Setup time
- Time between failures/ Repair time
- Transport time
- Conveyor length and speed

There are several common measures of performance obtained from a manufacturing simulator, including

- Throughput and throughput time
- Lead time or delivery dates
- Work in progress inventory level
- Resource utilisation, resource down time, preventive maintenance time
- Proportion of parts reworked or scrapped
- Time parts spend on waiting for a transport system, waiting for opertator or blocked.

Most of these performance measures can be directly taken from simlation outputs.
To summarise, the main purpose of the simulator is to generate outputs of a system for a given set of input variables.

6.2.2 Choice of Simulation Package

A generic manufacturing simulator is required to have the basic elements of manufacturing systems (with some space for enhancements) and a platform to accommodate their interactions. They include the simulation kernel (the main engine), layout modeller (physical elements - mostly stationary), entity flow modeller (raw material arrival details, operation details, routing, etc), and interfaces (user interface, computational modules ,etc.).

Application oriented specialist simulation packages have merits over general purpose simulation packages in that they are easy to use and are fast in model building. There are many specialist simulation software packages on the market, the majority of which can be described as visual interactive modelling systems (VIMS) (Robinson, 2004). VIMS enable a simulation to be built and run in a visual and interactive manner. The software provides a predefined set of simulation objects (designer elements). The modeller selects the required objects and defines the logic of the model through a series of menus. The visual display is also developed through a set of menus, thus the modeller requires little in the way of programming skills, although most of VIMSs either link to a programming language or have their own internal language to enable the modelling of complex logic.

The specialist simulation packages for manufacturing applications include (Robinson 2004; Law and Kelton 2000):

- Witness,	Lanner Group, Inc
- Arena,	Rockwell software
- AutoMod,	Brooks-PRI Automation
- Enterprise Dynamics,	Incontrol Interprise Dynamics
- Extend,	Imagine that, Inc
- ProModel,	Promodel Corporation
- QUEST,	DELMIA Corporation

All these specialist simulation packages can handle the modelling of manufacturing systems with a reasonable degree of flexibility.

Witness has been chosen as the main simulation package for the project. Among the reasons for its suitability are:

- it is available at Anglia Ruskin University and the University of Bath;
- it contains all the required manufacturing shop floor elements for modelling. It has in-built manufacturing constructs for model building ;
- it has in-built features that can accommodate sub-programs, giving further flexibility in modelling;
- it is compatible with the windows environment;
- it can be controlled by external computer programmes
- it is one of the most widely used simulation packages in industry (Swain, 2006)

6.2.3 Basic Features of Witness

The Witness package is capable of modelling a variety of discrete manufacturing elements. Depending on the type of the element, each can be in a number of "states". These states can be idle (waiting), busy (processing), blocked, in-setup, broken down, or waiting for labour, setup or repair. The most basic discrete modelling elements in Witness *are Parts, Buffers, Machines, and Conveyors.*

Parts are objects that flow from one location to another. They may be pulled passively into the model by the simulation, pushed into the system by an active part arrival schedule, arrive from a part file, be created via a production machine, or any combination of the above.

Buffers are passive storage areas of finite capacity. Buffers can be configured as "delay" buffers where parts may stay in for a minimum amount of time, or they can be configured as "dwell" buffers, where they cannot stay in the buffer any longer than a specified time. A part can be optionally ejected from a buffer if it violates any of the conditions. Combinations of priority dispatching rules are possible, as well as the ability to have parts pushed to and pulled from locations in the buffer.

Machines are the workhorses of Witness. A variety of machine types are available: Single, Batch, Assembly, Production, Multiple-cycle, and Multiple-station. Machines can be defined with Setup and Breakdown parameters which are useful for modelling real-life failures, retooling, preventive maintenance, etc.

Conveyors are defined by a length in parts and an index time which represents the time it takes a part to move from one position on the conveyor to the next. Parts can be pushed to and pulled from any position on the conveyor. The conveyor itself can actively pull and push parts. Conveyors could be fixed or queued. A fixed conveyor maintains the space between parts if the part on the front of the conveyor is blocked. By contrast, a queuing conveyor allows parts to compact together even though the conveyor may be stopped. The only time a queuing conveyor stops is when there are no gaps left, it is completely full, and no parts are being removed from it.

Other discrete elements of Witness include tracks and vehicles, labour, shifts, variables and attributes. The main elements of Witness for discrete simulation are shown in Figure 6.2.3.1.

The Witness user interface is Windows compliant. The primary interface is either pull down menus or tool bars. The operation of the simulation model is controlled from a toolbar usually at the bottom of the screen. The majority of the activities, however, take place in the simulation window. It is in this window that elements are placed in drag-and–drop fashion. Pre-constructed designer elements are commonly used to quickly drag-and-drop pre-defined items into the simulation screen. After the required elements are on screen, push and pull rules are added via the mouse visually.

Once the basic model has been built on the screen, the next step is to add more detail to the elements in the model. The details are entered via a *detail dialogue* box invoked by double left clicking the element. All of the logic for an element is entered from this detail dialog. A typical Witness layout with some elements, design elements window, and a detail dialog box is shown in Figure 6.2.3.2.

WITNESS Elements

Discrete	**Logical**	**Graphical**
Parts	Attributes	Pie charts
Buffers	Variables	Time series
Machines	Distribution	Histograms
Conveyors	Files	
Labour	Functions	
Vehicles	Part Files	
Tracks	Shifts	
Modules		

Figure 6.2.3.1. Main elements of Witness for discrete simulation

While the detail dialog controls the logic of the model element, the *display dialog* controls the look of the element. From it, any visual aspect of the modelling element can be controlled such as the icon to be used, text style and size, colour, and an assortment of other display items that can be added. The display dialog is invoked by double right clicking. Witness has a large number of pre-drawn icons which can be used to represent elements on the screen. In addition, a screen editor allows the user to add text and other graphics to a display. Bitmap files can be imported to Witness models as icons, and AutoCAD files can be imported to provide a shop floor layout to be used as a backdrop for a model.

Standard Witness reports can be viewed on the screen either in tabular or graphical format. In addition, several graphical elements are available for summarising statistics from a model such as pie charts, time series, histograms.

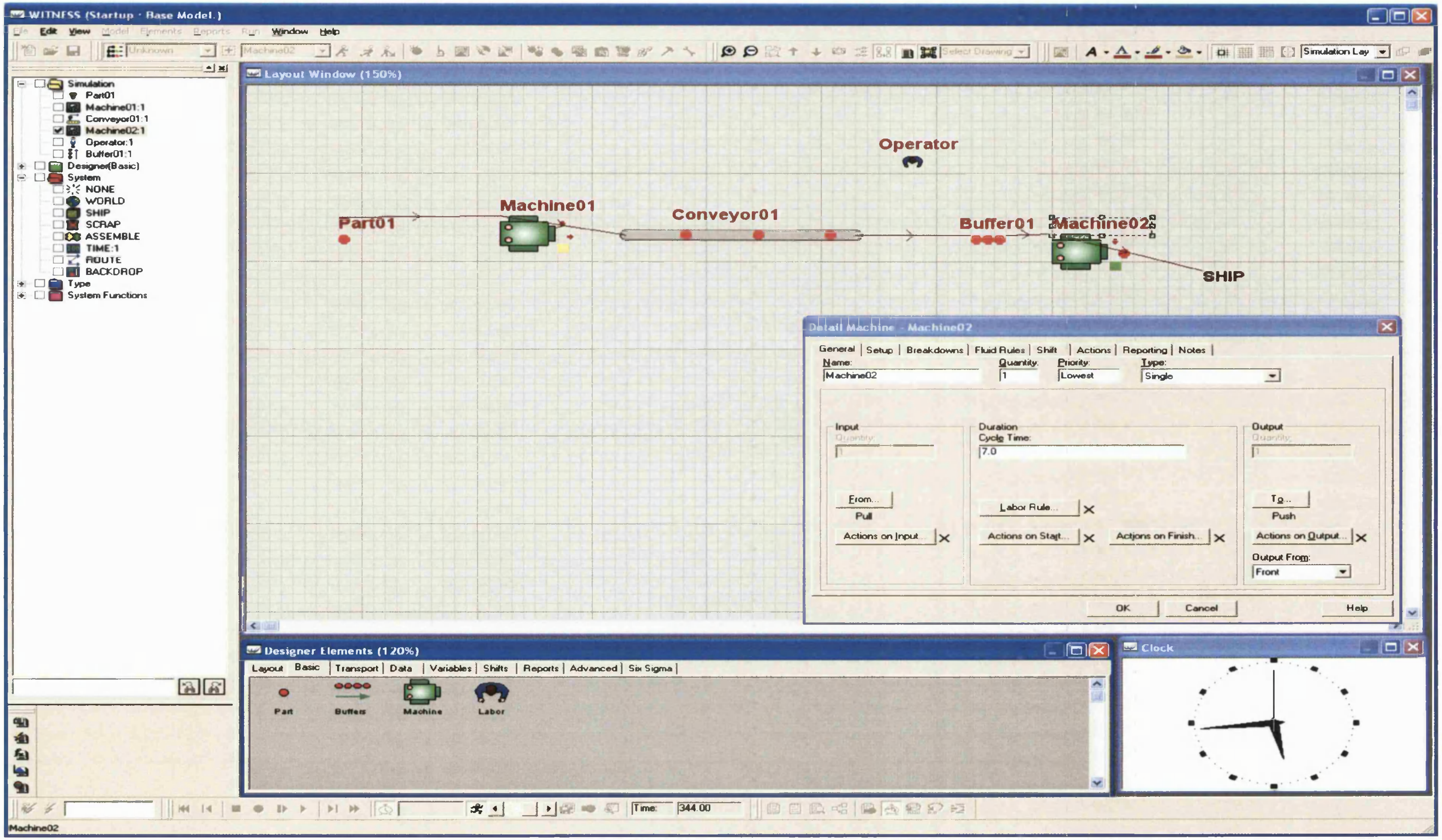


Figure 6.2.3.2. Witness simulation window with a detail dialogue box for a machine.

6.3 Data Consistency Check Element

It has been discussed in section 4.2 and 4.4 that control charts are the most common ways of checking data consistency. Among the known control charts, the cusum chart was shown to be the most beneficial in terms of detecting small changes in output data and in showing trends. Therefore, the cusum method is chosen to be the main tool for checking output data consistency. An acceptable size of the output data from the simulator would be tested with a cusum chart, within an acceptable level of confidence, whether the output data generated under different random seed streams are under statistical control or not. The way the data consistency check is conducted is shown in detail in section 7.4 of chapter 7.

6.4 Optimiser Element

The optimiser element would receive outputs from the simulator, assess system performance, and based on operational rules embedded in it, would effect changes to the input variables to enhance performance in the subsequent simulation run. The optimiser element is the main part of the research project.

As discussed in sections 4.2 and 4.3, unlike most of the simulation optimisation systems in the literature which are predominantly based on metaheuristic search methods, this optimiser is to be based on proven performance enhancing operations management methods such as the Theory of Constraints and line balancing (capacity and flow type). This approach avoids the need to run initial inferior settings and in the majority of cases leads to improved performance in each run. The optimiser is based on embedded optimisation rules, and will thus be referred to as an *Expert mechanism* (EM) from here on. The expert mechanism is explained in more detail in the chapter 7.

CHAPTER 7
EXPERT MECHANISM

This chapter discusses the rear-end elements of the integrated simulator-optimiser system: the Expert mechanism which is the optimising element, and the consistency checking element that uses the cusum method. The basic principles of integrating the Expert mechanism with the manufacturing simulator and the way the integrated system is developed are discussed in detail. It also covers an overview of some proven manufacturing performance enhancing methods that are used in this research to demonstrate the research concept.

7.1 Basic principles of Integrating the Expert Mechanism with a Simulator

In order to integrate a simulator with an expert system, it is necessary to provide a formal definition of each. This will enable those aspects of both components to be identified which are necessary for the interaction of both components without regard to the precise model used in each (Russel et al, 1994).

A simulator ***S***, as described by those aspects of the system that can be readily observed, can be defined as

$$S = \{A, K, O, T\} \qquad (7.1)$$

Where A = State variables (layout modeller and entity modeller)

K = The simulation Kernel

O = Simulation output parameters

T = Time of observation

Similarly, a knowledge-based expert mechanism ***E*** can be defined as

$$E = \{E_p, E_r, I, C\} \qquad (7.2)$$

Where, E_p = Set of data that describe the system (Input data)

E_r = Set of rules and relationships that transform E_p to C

I = Interface mechanism

C = Set of conclusions from E_p and E_r

For E to make decisions concerning S, then

- $V \subseteq \{A \cup O\}$ exists in S which is necessary and sufficient for making decisions
- E can use V consistent with E_p which is used in conjunction with E_r to derive C
- C can be used as input to S to effect changes in S for the next simulation run.

7.2 Creating an Integrated System

Creating the optimising mechanism requires that the environment it is developed in, is able to communicate with the simulator. With Witness as the chosen simulation system, one way of developing the mechanism is by using a program that can easily integrate with Witness. Witness can be controlled by other programs, that is, it is an object linked embedding (OLE) automation server, and Visual Basic is one of the OLE automation controllers that can control Witness. Therefore, it is reasonable to decide to develop the rule-based optimising mechanism in Visual Basic. This simplifies the process of interfacing E with S.

Relevant input/output values and the running of Witness could be controlled with Visual Basic with minor support from Excel. Visual basic being able to control both Witness and Excel is an ideal language to use for developing the expert mechanism. A simplified diagram depicting the system is shown in Figure 7.2.

Input data (state variables), A, for cycle time, set-up time, machine break down, buffer size, conveyor details, etc. would be displayed in Excel but controlled by Visual basic. The simulator uses these input data, runs the model and generates simulated results, O. The expert mechanism receives the relevant output data from the simulator, rearranges the data (with some help from Excel) giving Ep, assesses model performance (using Er) and generates recommended changes on the next set of state variables used as input data (C) to the simulator. The iteration goes on until a limiting factor is reached, at which point the results would be output. The system works with set constraints and displays the full trend within the limits, thus giving the user the opportunity to select a timely suitable model.

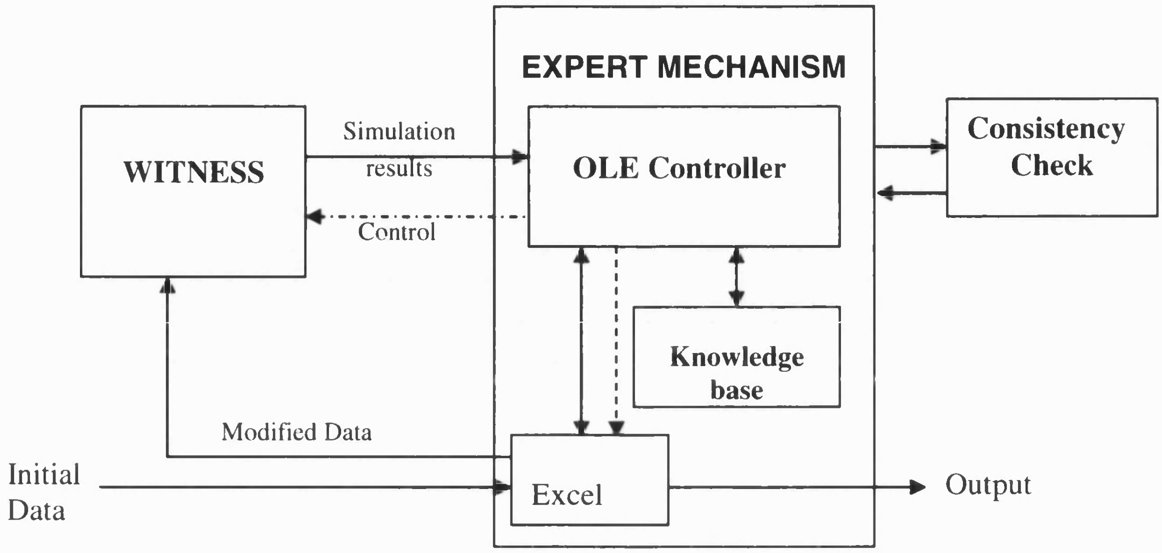


Figure 7.2. Integrated Simulator-Expert Mechanism Model

The data consistency check process could also be controlled by a subroutine within the Expert mechanism. The set of data output from the simulator, the mathematical manipulations and graphical display can be done in Excel with the control of the subroutine written in Visual Basic. More detailed account is given in section 7.4.

7.3 Manufacturing Performance Enhancing Methods

In this section some of the manufacturing performance enhancing methods that have been used to demonstrate the research concept have been described. Particularly, the theory of constraints and the principles of line balancing have been applied in the case studies presented in chapters 8, 9 and 10.

7.3.1 The Theory of Constraints (TOC)

The theory of constraints is all about exploiting the system constraints or bottlenecks. A system constraint as described by Goldratt (Goldratt and Cox, 2000) is nothing more than what we feel to be expressed by these words: anything that limits a system from achieving higher performance versus its goal. In reality any system has very few constraints, and at the same time any system in reality must have at least one constraint. In view of a system trying to improve, the general process of the theory of constraints can be summarised in five steps (Vonderembse and White, 2004B; Goldratt and Cox, 2000). They are:

- Identify the system constraints.
- Decide how to exploit the systems constraints.
- Subordinate everything else to the above decision.
- Elevate the system's constraints.
- If in the previous steps a constraint has been broken, go back to step 1, but do not allow inertia to cause a system constraint.

The theory of constraints is often linked to a chain analogy. Any chain has a weakest link. An increase in the strength of the chain can only be achieved by strengthening the weakest link. It makes no sense to strengthen an already strong link. Therefore first the weakest link (the constraint) in the chain (manufacturing system) is identified and then, it is made sure that the weakest link is not overloaded, by ensuring that the constraint sets the pace for the whole system. Next, effort is concentrated on improving the weakest link. With continual improvement, the weakest link will then be stronger than one of the other links. It will be futile to continue to improve the link further, so it will be necessary to identify and repeat the process on the new weakest link.

The theory, with re-analysed priorities of the manufacturing operation, can be briefly outlined as:

1. There is one operation in any business that prevents it from selling more product and making more profit. This is known as the bottleneck.
2. Exploiting this bottleneck by improved efficiency or similar means will make more money.
3. The product pushed through the bottleneck must be for sale and not for stock.

Improvements in bottleneck efficiency may be achieved by:

- Preventing the bottleneck waiting for work by having a buffer store just before the bottleneck operation.

- Reducing the scrap material processed by the bottleneck.
- Improved scheduling.
- Using the constraint income method to evaluate the profitability of orders.

Goldratt (1990) codified this theory into three rules for maximum profitability:

1. Maximise throughput, which is the rate at which the system turns inventory into product for sale.
2. Minimise operating expenses.
3. Minimise inventory.

If a business can improve with respect to all of these measures then the profitability of the business will improve in real terms and not just be a short term accountants view that may not be sustained in the long term. The other benefit of these measures are that they can be monitored daily and used to run the business.

Inventory

Like Just in Time (JIT), TOC stresses the importance of stock control and minimising inventory. However JIT wants to eliminate all work in progress that is not actually being worked on while TOC wants to eliminate all work in progress except the buffer store that protects the bottleneck. This can be looked on as an insurance policy it protects the bottleneck from unforeseen problems in material supply prior to that operation and like any insurance policy there is a premium in this case in the form of working capital tied up in the store.

The size of the store should be carefully calculated to give a service level of 95%. If the buffer operates at around 50% full then the system is operating correctly. If it runs nearly full then the system is very reliable and the size of the buffer can be reduced. If the buffer runs nearly empty most of the time then the supply system should be investigated to determine why the supply system is so unreliable.

Any investigation of the supply system should be carried out by recording the time and date of each operation prior to the bottleneck, the result then compiled to show where the delays are occurring and actions taken to eliminate these delays, permanently.

Schedule

Scheduling is a key part of the TOC system and must be constructed to:

1. give the customer what they want when they want it and this may require some negotiation to determine exactly what they want, delivery dates, batch size, etc.
2. minimise the inventory.
3. enable material to be released into the system in order to comply with delivery dates.
4. enable material to only be released into the system when it can progress through the system without any waiting, except at the buffer prior to the bottleneck (often called 'a line of sight through the system'). The buffer should be treated as a dummy operation in the simulation.
5. enable batches of work to be split for transportation if not for manufacture to minimise work in progress.
6. Constantly improve the system to further reduce inventory.

Operating Expenses

Operating expenses are the costs a factory must pay to turn inventory into products for sale and include:

- Wages
- Rent
- Power
- Indirect materials like tools etc.

Management must be constantly striving to reduce these expenses to ensure the business remains competitive. However the operating expenses must be viewed as a whole and the reduction of one part must not be allowed to increase the total, for example the sub-contracting of a category of work.

Constraint Income

This is a method of evaluating the effectiveness of the management decision making process. It involves evaluating the profit made when selling a product on the basis of minutes of bottleneck time. Thus if a product (or service) yields a gross profit of £10 and takes 10 minutes to process at the bottleneck it yields a constraint income of £1 per minute. If another product gives a gross profit of £5 and takes 2 minutes at the bottleneck this results in a constraint income of £2.50 per minute and should be given a higher priority as it is more profitable, providing the volume can be sold.

It can therefore be assumed that all the costs are borne by the bottleneck and all the income generated by it, as the rest of the business is there just to service the bottleneck.

Constraint Decision Making

Using the constraint income, decision making process can be extended to wider areas than just bottlenecks. If the closing of a line or factory is being considered then the following questions must be answered:

1. What will be the effect on throughput in £?
2. How much will the operating expenses be reduced?
3. How many people will be laid off, including those in service departments like wages etc.?
4. Is this change still economic?

The benefit of using this method is that it highlights the fact that the fixed overheads must still be paid even if a part of a factory is not used.

Process Improvement

The constraint income method can also be used to justify expenditure on process improvement. If for example, a proposal for a small improvement in a process that saves 10 minutes a week and the constraint income can be sold for £5 per minute then the improvement will increase profit by £50 per week. If that minor improvement costs only £200 then a 4 week pay back is achieved.

The benefit of this type of project evaluation is that it encourages the concept of continuous improvement which is a common feature of most modern management theories. However TOC has one important advantage over the theories that call for a general improvement strategy, it focuses these improvements where they will do most good and increase profits.

7.3.2 Line Balancing

Line balancing seeks to reduce the waiting time caused by unbalanced production times between individual work centres and ensures that production is predictable in quantities and timing. The techniques of line balancing include:

Flow Balancing: a type of line balancing where the flow is adjusted in such a way that WIP inventory will be kept low i.e, the faster machines are allowed to produce only the amount that can be handled by the bottleneck station. The bottleneck station determines the throughput.

Capacity Balancing: a type of line balancing where slower machines are either made faster (by adding resources or using better capacity resources) or made to work extra over-time hours to compensate the deficit they have compared to faster stations (bypass balancing). The latter is expected to raise the WIP. Capacity balancing is usually dictated by the required throughput.

Split and Group Balancing: Here the throughput (or cycle time) is usually fixed and resource utilisation is needed to be high. The operations are broken into possible smaller ones and then regrouped to workstations for better resource (usually human operators) utilisation. Often this balancing is applied to reduce labour cost in assembly lines and often requires a redesign of layout. In simple terms, if a fully balanced workload is achieved, then the idle time at each workstation will be zero (Vonderembse and White, 2004C). Mathematically, the number of workstations can be represented as

$$N = \frac{\sum_{i}^{n} ti}{C} \tag{7.3}$$

Where t_i = task time for element i, where i=1,2…,n

N= minimum number of workstations rounded to the nearest whole number

C = the required cycle time for the balance

Σ t_i = the total task time required to assemble one unit

Apart from the split and group balancing, other balancing types are usually inherent in the TOC procedure.

7.3.3 Priority Dispatching Rules

In job shop production and some batch production, the product mix affects the dynamics of shop floor processing. The central feature of modelling, therefore, would be the choice of the queue discipline. The schedule that is ultimately obtained and its merit judged on various criteria, is primarily determined by the manner in which jobs are selected from queues and dispatched through the shop. Here proper use of priority dispatching rules would help to improve performance.

In priority dispatching rules, the performance measures are assessed firstly by their effectiveness in getting their due dates and secondly their ability at reducing congestion and queuing time while maximising facility utilisation (Heizer and Render, 2005C). Typical priority dispatching rules include

EDD -	Earliest due date
FCFS -	First Come First Serve
SPT -	Shortest Process Time first
LPT -	Longest Process Time first
SLACK -	Operation with lowest slack time first
S/OPN -	Operation with lowest ratio of Slack time to number of operations first
OPNDD-	Operation with earliest operation due date first
SET -	Minimum setting time first
CR -	Critical ratio = time remaining to due date/ process time remaining

The limitations of dispatching rules is the fact that scheduling is dynamic and the rules have to be revised to adjust to changes in process, equipment, product mix, etc. The rules may not help in recognising the bottleneck resources in other work stations or departments. Therefore, it is difficult to decide on the rule or rules that will lead to an overall improvement in performance without conducting a considerable size of experiments.

7.4 How the Integrated Simulator-Expert Mechanism System is Developed

7.4.1 Summary

The program of the Expert Mechanism has the following main components:

1. Linking the Visual Basic (VB) program with Witness
2. Launching Witness
3. Loading a witness model
4. Downloading a list of model elements to Excel
5. Loading initial input data to the Witness model from an Excel control page
6. Specifying the run time and Running the Witness model
7. Conducting an initial consistency check using Cusum

If 7 is ok, the optimising performance is started,

8. Generating simulation outputs
9. Assessing performance using the simulation results
10. Checking external constraints
11. Identifying bottlenecks and effecting changes to enhance performance
12. Conducting a final consistency check

A summary outline of the integrated system is shown in Figure 7.4.1.

To give the reader a feel of the program elements, the main steps followed in each component of the expert mechanism are described in the following sub sections. References are also made to appropriate sections where detail VB codes can be seen.

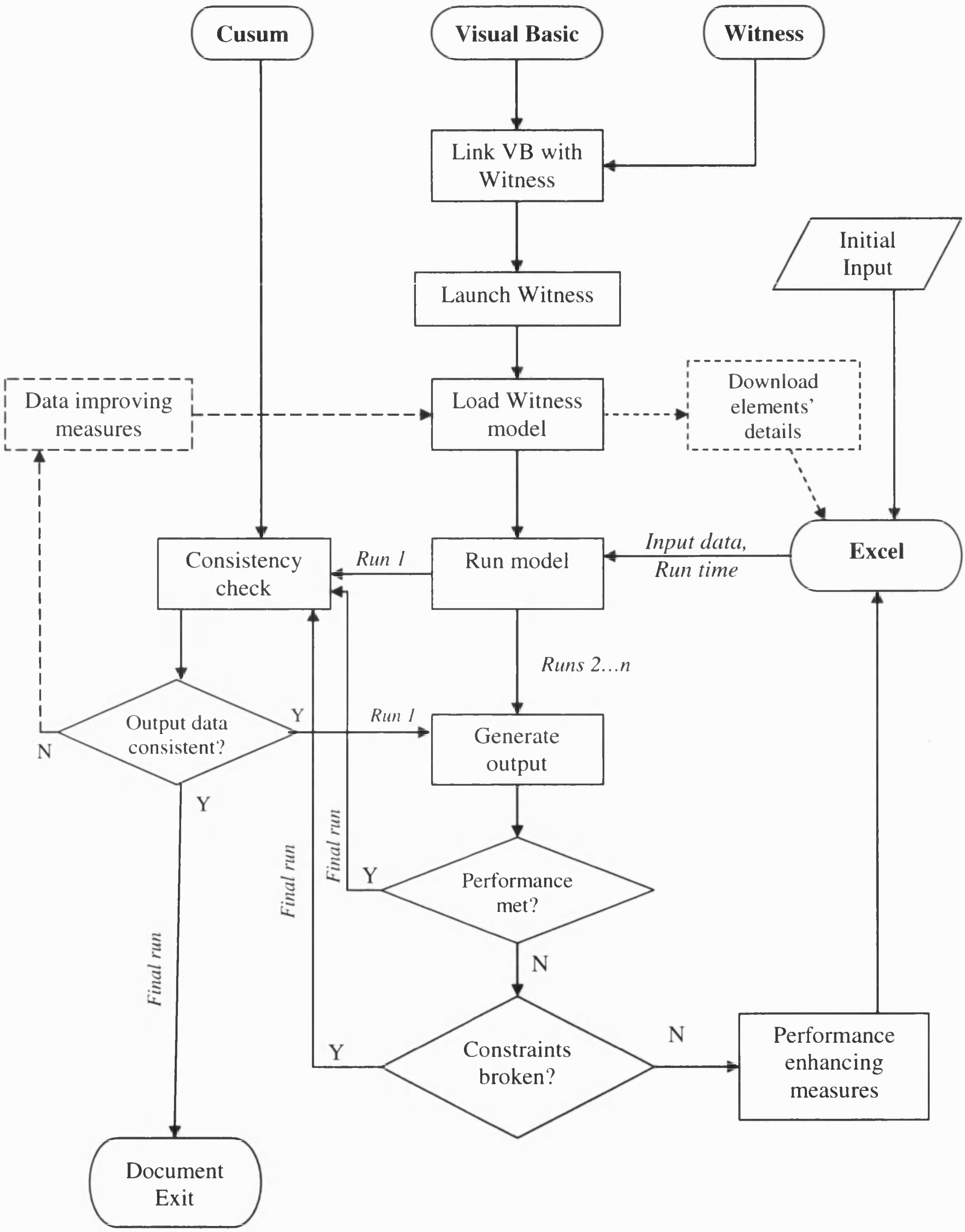


Figure 7.4.1 Summary outline of the Simulator - Expert mechanism system

7.4.2 Details of Expert Mechanism Components

1. Linking Visual Basic to Witness:

A running Witness model as an OLE automation Server is linked to Visual Basic (VB) by creating an object in the VB program. The name of the object in our Expert Mechanism program is ***gobjWitness*** and the link to Witness is created as

Option Explicit

Dim gobjWitness As Object

2. Launching Witness:

Launching of Witness from the VB environment has the following main steps

- creating a VB function and relevant variables
- stating the directory to launch Witness from
- Checking whether Witness is open or not
- If open, connect it to the VB
- If not open, kick off Witness and link to VB

The visual basic program that does the Witness Launching is shown as a function in Appendix B under the "Public Function OpenWitness()" subroutine.

3. Loading the Witness Model:

Once Witness is opened and connected to VB, the loading of the model follows the following main steps:

- creating a VB function with the model name as a variable
- referencing the model name variable to an excel entry
- activating the OLE link
- loading the model into Witness from the excel cell.

For a detailed visual basic codes on loading the Witness model, please see the subroutine "Public Function LoadWitnessModel" in Appendix B.

4. Downloading list of model elements to Excel

This part is optional. If the list of model elements and their relevant input variables are already in Excel, there is no need to run this function. In the cases where the Witness model is built without any connection to Excel, the model element names and relevant variable data are downloaded to the Excel control page. As an example, the following simplified visual basic procedure shows how the element names are downloaded into the control page. A typical control page is shown in Table 7.4.2

```
Sub LoadElementsToExcel()
Dim i As Integer, k As Integer
Dim element As String
For i = 1 To gobjWitness.Count
        'gobjWitness.Count is the total number of elements in the witness model
  k = 10 + i                      ' This counter starts from 11
  element = gobjWitness.Name(i)    ' Assigns the name of the element
  Sheets("Control").Cells(k, 1) = "CycT_or_Cap(" & i & ")"  'Optional variable name
  Sheets("Control").Cells(k, 3) = element    ' Puts elements starting from row 11

 Next i
End Sub
```

5. Loading initial input data to the Witness model from an Excel control page

The relevant input data mainly cycle time, buffer size, conveyor speed and labour quantity are loaded to the Witness model automatically from the Excel control page before each run. The procedure followed for loading relevant data of four element types is summarised as follows:

Using a "For" loop for each element, if the element type is

- "machine", assign the specified cell value for the cycle time of the machine
- "buffer", assign the specified cell value for the capacity of the buffer
- "conveyor", assign the specified cell value for the index time of the conveyor
- "labour", assign the specified cell value for the quantity of the labour.

For a more detailed VB program on loading input data please see the subroutine "TransferDataToWitness" in Appendix B.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2	WITNESS folder	C:\Program Files\Witness 2004 Release 2						Run Flywheel			ConsTest Initial		ConsTest_ Last			Reset	
3	WITNESS model name	F:\Habtom's files\Research\flywheel4.mod															
4	Model run time	2840.0															
5				Total runs	14												
6	RESULTS																
7	Run #				1	2	3	4	5	6	7	8	9	10	11	12	13
8	Parts shipped				2323	2411	2491	2531	2628	2700	2767	2815	2930	2988	3071	3199	3272
9																	
10	**Model Data:**	Original data		Modified Data	Work hour												
11	CycT_or_Cap (1)		Fork_raw														
12	CycT_or_Cap (2)		Trunion_raw														
13	CycT_or_Cap (3)		Turn		40	40	40	40	40	40	40	40	40	40	40	40	40
14	CycT_or_Cap (4)		Mill		40	40	40	40	40	40	40	40	40	40	40	40	40
15	CycT_or_Cap (5)	5	RoughturnOld	5	40	40	40	40	40	40	40	40	40	40	40	40	40
16	CycT_or_Cap (6)	1	RoughturnNew	0.584679289	40	40	40	42	44	46	48	50	52	54	56	58	60
17	CycT_or_Cap (7)	1.2	R_turnNew2	0.606081544	40	42	44	46	48	50	52	54	56	58	60	62	64
18	CycT_or_Cap (8)	0.55	Finebore	0.55	40	40	40	40	40	40	40	40	40	40	40	40	40
19	CycT_or_Cap (9)	0.7	Drilltap	0.353547567	40	42	44	46	48	50	52	54	56	58	60	62	64
20	CycT_or_Cap (10)	0.75	Hassy	0.559661547	40	40	40	42	44	44	46	46	48	48	50	52	52
21	CycT_or_Cap (11)	0.25	Grind	0.25	40	40	40	40	40	40	40	40	40	40	40	40	40
22	CycT_or_Cap (12)	1.2	BoreTurn	1.2	40	40	40	40	40	40	40	40	40	40	40	40	40
23	CycT_or_Cap (13)	24	Hob	18.804628	40	40	40	40	40	40	42	42	44	46	46	48	50
24	CycT_or_Cap (14)	0.75	Chamfer	0.533010998	40	40	40	42	42	44	46	46	48	48	50	52	52
25	CycT_or_Cap (15)	1.2	Fineturn	1.088435374	40	40	40	40	40	40	40	40	40	40	40	42	44
26	CycT_or_Cap (16)	0.7	Balance	0.548468317	40	40	40	40	40	42	42	44	46	46	46	48	50
27	CycT_or_Cap (17)	0.4	Pack	0.4	40	40	40	40	40	40	40	40	40	40	40	40	40
28		Total work_hour/week			520	524	528	538	546	556	568	576	590	598	608	624	636
29		Op_cost-flat (£)			9909.70	10024.98	10135.05	10306.24	10485.44	10677.43	10895.20	11042.55	11320.56	11474.41	11673.55	11989.05	12210.73
30		Cost/part (£)			4.27	4.16	4.07	4.07	3.99	3.95	3.94	3.92	3.86	3.84	3.80	3.75	3.73
31		Op_cost-mod (£)			9909.70	12951.06	13083.47	13310.50	13534.38	13782.20	14066.99	14259.01	14615.20	14813.73	15068.70	15473.55	15762.23
32		Cost/part (£)			4.27	5.37	5.25	5.26	5.15	5.10	5.08	5.07	4.99	4.96	4.91	4.84	4.82
33																	
34	Notes:																
35	Machine states:		Model Status:				Fixed cost/week, Cf (£)				9000						
36	0	Off-shift	1	Running			Overhead cost/week, Cov (£)				3600						
37	1	Idle, waiting p.	2	Stopped, waiting for input			Labour cost/week, Cla (£)				1800						
38	2	Busy	3	Other state(e.g. error message)			Material cost/part, Cma (£)				3						
39	3	Blocked	4	Stopped b/c of witness error													
40	4	Setting up															
41	5	Down and repair			Operating Cost(flat), Cop =					Cf + (Cov + Cla)*(Wh/Who) + Cma * X							

Control / Costing / Sheet1 / Sorted_data / ConsTest / sheet3 / Sheet4 / Sheet5

Table 7.4.2 A Typical Control Page.

6. Running of the Witness Model

To run the witness model, a run time value is picked from the Excel control page and then, with the control of the VB, the model is run to the specified run time either in a *run* mode or in *batch* mode. The VB section for running the model can be seen under the subroutine "RunWitness" in Appendix B.

7. Conducting consistency check using Cusum

To check the consistency of the simulation output data, simulation results with different random stream seeds have to be tested using the cusum chart. Initially a separate unique whole random stream number + a variable (RStream) is assigned to each stochastic input data in the model. Then the value of the variable is automatically changed in each subsequent run using a counter. As an example, if the cycle time of a machine has a normal distribution with a mean of 10 minutes and a standard deviation of 0.1 minutes, the format in witness will be as follows:

NORMAL(10,0.1, 3 + RStream)

where 3 + RStream is the random stream number assigned for that machine.

For the first run, RStream will have a value of 0. But in the subsequent runs the program in VB increments the RStream value in steps resulting in a different random stream value. The results of the simulation runs for 25 different random number stream settings are then obtained and displayed in Excel. The data are then tested by a cusum chart for consistency, as explained in chapter 4.

In summary, the steps are

i. Create a variable called RStream
ii. Add this variable to every random stream number in each stochastic input data
iii. Run model from the Visual Basic environment (Expert Mechanism)
iv. Obtain output data and put in a specified Excel cell
v. Increment RStream by a step
vi. Repeat steps 3 to 5 until RStream = 24
vii. Calculate target value and standard deviation from the 25 output data
viii. Calculate the cusum values
ix. Call the cusum control factors
x. Plot the upper and lower masks
xi. Plot the cusum values, the upper mask and the lower mask
xii. Check if the cusum chart is within the upper and lower masks.

This cusum checking can be done automatically from the Expert Mechanism. For the experimental purpose, the whole process is completed by pressing a VB control button as shown at the top of table 7.3.1. For more details in the VB program please see the subroutine "ConsTest_Ini" in Appendix B.

8. Generating Simulation Outputs

The relevant simulation outputs are obtained using counters in VB. The typical outputs: parts output (Final parts), average work in progress (Av.WIP), quantity of element , percentage utilisation of resources, percentage setup time, percentage repair time, percentage idle, percentage blocked, useful percentages (Util & Setup) are shown in Table 7.3.2

	A	B	C	D	E	F	G	H	I	J	K
1											
2	**Final parts**	**3321**		**AV.WIP**	**6.0948757**						
3	**Sequence No.**	**Element Nam**	**Qty**	**Utilisation**	**Setup**	**Down Repair**	**Blocked**	**Idle**	**Util & Set-up**	**Elem.type**	Total
4	1	FlywheelP	0							1	0
5	2	FlywheelD	0							1	0
6	3	StarterringP	0							1	0
7	4	StarterringD	0							1	0
8	5	RoughturnOld	4	22.42067202	2.817384	0.65719075	73.8981	0	25.23805603	2	99.79
9	6	RoughturnNew	1	77.13614596	3.9967575	1.44274618	17.41308	0	81.13290342	2	99.99
10	7	R_TurnNew2	1	79.98679499	6.1632473	9.30831437	0.858458	3.6783	86.15004232	2	100
11	8	Finebore	1	9.553588381	3.4700855	0	85.52736	0	13.02367385	2	98.55
12	9	Drilltap	1	52.76697954	35.107425	2.47505236	3.969035	2.4713	87.8744046	2	96.79
13	10	HTassy	1	79.70436751	0	0	4.250755	13.685	79.70436751	2	97.64
14	11	Grind	1	34.90384616	0	2.09076492	59.53224	0	34.90384616	2	96.53
15	12	BoreTurn	1	27.69230879	4.8323424	0.95662307	64.4209	0.5767	32.52465114	2	98.48
16	13	Hob	2	72.7000069	5.7692308	0.8207915	20.48727	0	78.46923767	2	99.78
17	14	Chamfer	1	77.77476918	2.6361676	2.25983611	17.31645	0	80.41093673	2	99.99
18	15	Fineturn	1	77.44087637	0.7266452	2.59049872	3.203038	16.039	78.16752158	2	100
19	16	Balance	1	77.85691151	0.7241275	3.86278848	1.912776	15.641	78.58103904	2	100
20	17	Pack	1	56.78491489	0	0	0	40.111	56.78491489	2	96.9
21	18	Operator	5	59.06307434				40.937		5	
22	19	Buffer_P	1							4	
23	20	Buffer_D	1							4	

Table 7.4.3 Typical simulation output data

After each simulation run, the expert mechanism uses counters and name matching to obtain the output data and place them in the right location in the Excel output page. The main steps can be summarised as follows

i. Run the witness model
ii. Obtain the final parts from witness and put it in Excel cell(2,2)
iii. Obtain the average WIP from witness and put it in Excel cell(2,5)
iv. Set counter to 1 (same as sequence number in Table 7.3.2)
v. Obtain name of element (counter) from the Witness model
vi. Put name of element in the Excel cell (3 + counter, 2)
vii. Put quantity of element in the Excel cell (3 + counter, 3)
viii. Put percentage utilisation of element in the Excel cell (3 + counter, 4)
ix. Follow similar steps to put the rest of the data in other columns
x. Increment the counter by 1

xi. Repeat steps (v) to (x) until counter = total number of elements in the model

xii. End

For more detail in the way simulation results are obtained using the VB program, please see the subroutine "GetWitnessResults" in Appendix B.

9. Assessing performance

Depending on the chosen performance measure, the relevant output data along with the input data is then used to quantify performance. In this research, as the main demonstrator, improving throughput is the aim with justification elements added to it. Therefore, the final parts are compared with the required output. In instances where cost-benefit evaluations are needed, the calculations are done in Excel under the control of VB.

10. Checking external constraints

This is a simple check against set constraints before proceeding with the performance enhancing process. Examples include

- checking the final parts obtained from simulation against the maximum required output
- checking that the shift is consistent with allowed maximum number of hours
- checking that the cost incurred does not exceed the benefits
- checking that the efficiency of resources is realistic
- checking additional constraints such as against the maximum number of resources possible, etc.

Constraints are mainly controlled by "IF .. THEN" statements as:

```
IF (final parts < maximum_required_output) AND .... THEN
        Proceed with optimisation process
ELSE
        Do calculations, graphs, etc
        Display results as final ...
 ENDIF ...
```

11. Identifying bottlenecks and effecting changes to enhance performance

After the first run, the simulation output data are manipulated to show the most utilised elements at the top of an Excel sheet. This is done by a subroutine in VB which identifies the relevant output data, calculates the total effect (unavoidable utilisation) and sorts the table with the total effects in descending order so that the bottleneck (constraint) elements appear at the top. An example of a sorted data is shown in Table 7.3.3.

The VB subroutine now picks the constraint elements, and provided that conditions of external constraints are not broken, effects changes in variable values of the constraint elements for better performance.

	A	B	C	D	E	F	G	H	I	J
1	SORTED OUTPUT DATA									
2						Down			Utilisation	
3	Seq. No.	Element Name	Qty	Utilisation	Setup	Repair	Blocked	Idle	& Set-up	Elem. type
4	9	Drilltap	1	69.35331	23.81053	3.620585	1.842808	0.06182	93.16384	2
5	7	R_TurnNew2	1	84.25762	3.107052	10.71249	0	1.91594	87.36467	2
6	10	HTassy	1	74.42081	0	0	3.601222	21.4784	74.42081	2
7	14	Chamfer	1	71.69645	2.273776	1.714828	24.29785	0	73.97022	2
8	6	RoughturnNew1	1	70.21488	2.129239	1.442746	26.20847	0	72.34412	2
9	16	Balance	1	69.47209	0.339304	3.862788	0.329412	25.9964	69.81139	2
10	13	Hob	2	63.23802	3.942308	0.820791	31.94048	0	67.18033	2
11	15	Fineturn	1	59.53846	0.491448	1.276553	3.40125	35.2923	60.02991	2

Table 7.4.4 Manipulated Simulation Output Data

The variations of performance enhancing processes used in this research as models include

A. Improving throughput without incurring cost on resources. Here, within the existing resources, attention is given to the constraint elements whose input variable changes (mainly overtime work) can result in improved throughput.

B. Improving throughput by incurring cost to additional or improved resources. Variation to the variable values of the constraint elements (such as element quantity or element processing speed) is made to enhance performance, but at the same time cost-benefit analysis is conducted.

C. Reduction of cost for a given throughput. Here the sorting of the simulation outputs would be the opposite of 1 and 2 whereby the least utilised elements appear at the top of the table. A possibility of reducing cost on these non-constraint elements is then explored and changes effected.

D. A combination of the above.

In case A, the theory of constraints along with bypass balancing has been used as it is one approach that could be effective, provided the effect of work in progress is assumed insignificant. The constraint elements are identified and then allowed to work on overtime. This usually leads to enhanced throughput and generally better operating cost especially if the fixed cost is high. A case study model that demonstrates this approach is explained in more detail in Chapter 9. The VB program that automatically effects the performance optimisation process is shown under the subroutine "*Modelchanges*" in Appendix B. The overtime work is effected by a proportional adjustment to the processing time of relevant elements. The simple cost calculations are done in a linked Excel environment but controlled with VB.

In case B, the theory of constraints is again used to identify the constraint elements. Changes are then made to the constraint elements to enhance their processing speed such as reducing process time, reducing setup time, and adding new elements. As investment is usually associated with these changes, a cost justification process could be included. A case study model that demonstrates this scenario is explained in more detail in chapter 8. Costs for the possible changes are put in an Excel sheet and used by the VB program to do cost-benefit calculations.

In case C, the main aim is to reduce cost. This can either be done solely to reduce cost on an existing Witness model with satisfactory throughput value, or as an extension to an optimisation process to reduce cost further. The simulation output data manipulation follows the reverse of constraint element identification. The data are sorted based on the unavoidable utilisation in ascending order so that the most underutilised elements appear at the top of the table. The VB program of the Expert Mechanism then picks elements from the top and effects changes to reduce cost. One example is to reduce labour quantity were the utilisation is low. This would work as follows:

$$\text{IF } NQTY(Ei) * (1-Ui) > 1 \text{ THEN Set } NQTY(Ei) = NQTY(Ei) - 1 \qquad (7.4)$$

Where NQTY(Ei) = the quantity of element i

Ui = Unavoidable utilisation of element i

A more detailed demonstration is shown in chapter 10.

In case D, a combination of two or more of the options are applied with proper settings of conditional limits. A combination of case B and case C has been applied in the case study of Chapter 10.

12. Conducting final consistency check

This is similar to the initial consistency check except that it is done on the final model.

7.4.3 Generic Framework

The Expert Mechanism has a common set of subroutines that will use components 1 to 7 (Expert mechanism components are listed in section 7.4.1). These subroutines apply to all anticipated scenarios and are automatically included. At the moment, component 4 (downloading list of elements from Witness to Excel) and component 7 (conducting cusum check) are optionally applied and are effected by pressing a VB button, but they can easily be part of the automatic runs.

Components 8 to 11 are optimisation options that apply to different scenarios. Therefore, a choice among the possible options (such as described as A to D in section 11) is to be made by selecting a button and entering relevant data (such as constraints and costing) in specified Excel sheets. The Expert Mechanism would then use the relevant VB rules and conduct the optimisation process. In a fully developed system, instructions would be given to the user where to put the input data. For instance, if option A is chosen where TOC and bypass balancing are to be applied, the additional input data from the user could include constraints on maximum work time and maximum throughput, and fixed cost, overhead cost, material cost and labour cost. A predefined Excel sheet would have cells assigned for these data. The Expert Mechanism would then pick these data in each simulation run to do calculations, and to check that the constraint conditions have not been broken.

It should be understood that the expert mechanism is developed as a concept to support the research proposal. Especially on the performance enhancing element of the expert mechanism, TOC and line balancing (supported by cost analysis) were used as exemplars of proven performance enhancing methods in manufacturing environment. Their application fits to the objectives of the case studies (discussed in chapters 8, 9 and 10) and are used to illustrate the research concept.

CHAPTER 8
CASE STUDY ONE

(The results of this case study have been published in the Journal of Engineering Manufacture Proceedings, Vol 218 Part B, IMechE, February 2004, pp245-249. See Appendix A and F)

This chapter, upon completion of the stages of developing the Expert mechanism and integrating it with the manufacturing simulator, presents a first case study to demonstrate and, to an extent, validate the integrated simulation-Expert mechanism system.

8.1 Aim of the Case Study

The main aim of this case study is to demonstrate the use of an expert mechanism in an automatic interpretation of manufacturing simulation results and in effecting changes to the input data for optimising purposes. It also demonstrates how the expert mechanism and the manufacturing simulator are integrated.

8.2 Base Model

As the first step to demonstrate how simulation results can be translated into performance enhancing actions, a case study factory model with limited operation and resource flexibility has been set up as shown in Figure 8.2. The experiment was based on a company model obtained from the Lanner Group, the software house behind the Witness simulation modelling package. Some operational data have been modified for simplicity.

At the start of the case study, the simulated company experienced a severe backlog in sales orders due to antiquated machinery and poor production planning. Assuming that there was a demand for up to five times the current product output, a series of simulation runs were set up to evaluate the effect of investing appropriate resources against the possible increase in throughput and other benefits.

The model consists of seven main operations. The manufacturing process starts with the stock of bars that come into the saw area stock buffer. The bars are then cut producing 3 blocks from each bar. After sawing, the blocks go to a belt conveyor that transfers the cut bars to the coating operation. The coating machine coats six blocks at a time. Once coated the blocks are placed in the staging area adjacent to the inspection station. The inspectors then determine the quality of each block's coating and send it either to hardening, or to the rework buffer. The hardened blocks are then loaded into special fixtures so that four blocks can enter a grinder together. There are two grinders available with no priorities between them. Once ground, the fixture and the four blocks are placed into an unloading station where the blocks (now valves) are sent to the finished stock areas and the fixtures onto an overhead conveyor. The conveyor puts the fixtures back into the fixture buffer for reuse by the loading machine. Witness was used to model the system.

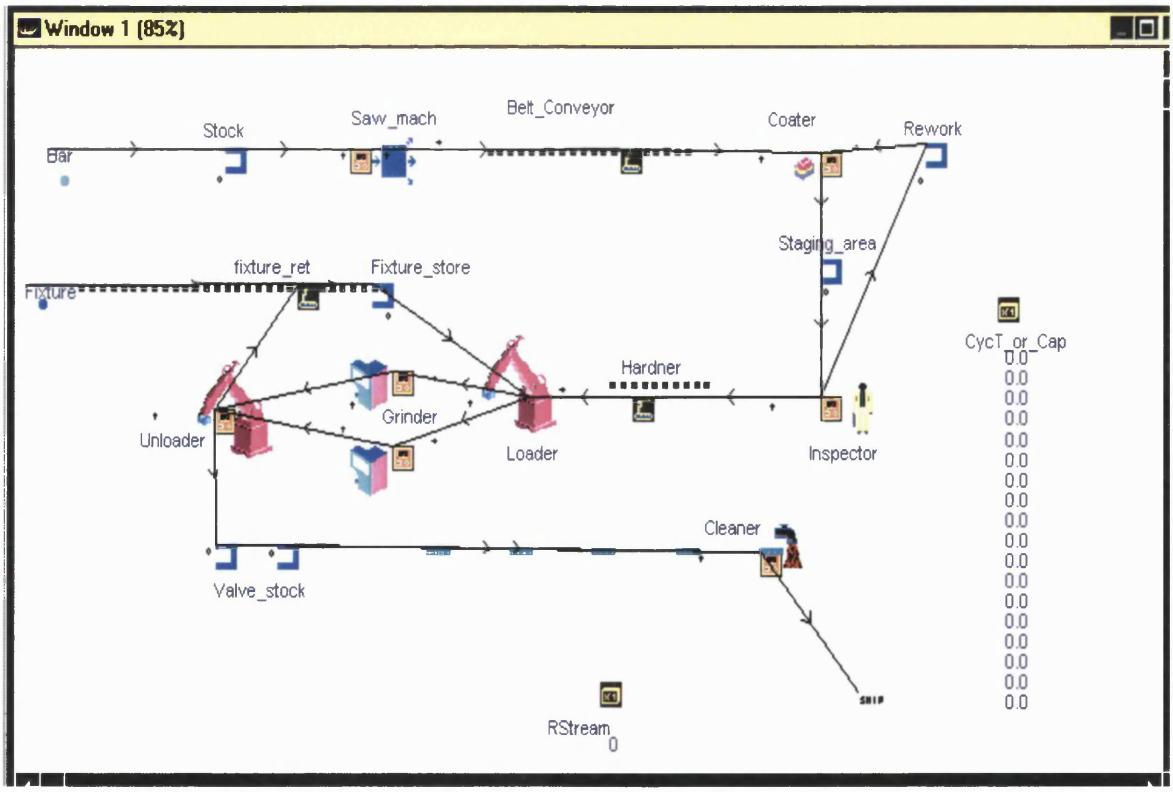


Figure 8.2. The simulation model used in Case Study 1.

The model uses stochastic elements especially on the machine mean time between failures (MTBF) and repair times. As an example, the saw machine has a MTBF which is of negative exponential distribution with the mean of 100 minutes i.e., NEGEXP(100), and a repair time which is of triangular distribution with the lower, mode and upper values of 10, 25 and 30 respectively, i.e, TRIANGLE (10,25,30).

8.3 Model Data Consistency Check

It was important to check that the model output variability was not statistically significant so that the optimisation process could proceed effectively. The data consistency of the model output data was checked using cusum chart, by conducting 25 runs with different random number streams for each run and for each element and each corresponding data. This was meant to give the feel that events were following more realistic randomness.

Different initial random number seeds were given to the stochastic data. A variable, named RStream, was created and added to the random stream of each data. Then the RStream was automatically changed in each run i of the 25 runs conducted. The recorded output values Yi of the 25 runs are shown in the table of Figure 8.3. The target value T which was the mean of the output values was then calculated. The x-T and the cusum values follow the next two rows.

The upper and lower masks of the cusum chart were calculated based on the control factors read from the table of C2 semi-parabolic mask as defined in BS 5703 part 3 (see Appendix D). The cusum chart (90% confidence) is also shown in Figure 8.3. It could be seen from the graph that the data were all between the upper mask and the lower mask, indicating good statistical consistency. A similar output data consistency check was conducted on the last model and it again showed satisfactory output data consistency. The cusum check was done automatically by pressing a VB control button.

8.4 The Rule-based Expert Mechanism

8.4.1 Case Study Objectives

The existing system can manufacture around 144 valves every 75 hours. It has been pre-established that the benefits of an increase in throughput by 1 valve can be fully justified for an investment of £250. That is, for each investment of £250, there must be an increase of at least one valve. A maximum amount of £75,000 is available for the investment, which amounts to an equivalent of 300 more valves to justify the total spending. The main investment costs expressed in terms of production benefits are shown in Table 8.4.1. Each item has been assigned a cost equivalent in parts.

Element	INVESTMENT COST-EQUIVALENT IN PARTS		
	New Element	Decrease cycle time by 10%	Decrease set-up time by 10%
Buffers	0.2		
Saw_machine	100	10	20
Conveyor		10	
Coating machine	80	24	
Inspector	64	24	
Hardener (Furnace)		24	
Loader/ Unloader	64		
Grinder	200	40	
Cleaner	40		

Table 8.4.1. Possible costs for Investment

8.4.2 Methodology

As previously described, the main performance index is net profit (or net saving) which is the difference between the increase in throughput and the investment (expressed in terms of equivalent parts). Similar to Goldratt's claim, increasing throughput is the main performance measure but should be justified for any costs incurred. The main rules used for enhancing the performance were based on the technique of Theory of Constraints (see section 7.3.1) backed by concurrent monitoring of investment. It involved mainly identifying bottlenecks and blockages which were then used as the basis for actions to be taken in each sequential simulation run.

As expressed in the last chapter, Witness is an object link embedding (OLE) automation server that can be controlled by Visual Basic (VB) which is an OLE controller (Lanner Group, 2004). Relevant input/output data to Witness as well as running of Witness could be controlled with VB (with some assistance from Excel). Therefore, rules that govern the data consistency check and performance enhancement are written in Visual Basic.

The main steps in conducting the optimisation process are:

Data Preparation (Control Table):

1. Reset control table
2. State the model file to launch (open)
3. State values for constraints
4. Download list of model elements Ei (Optional), where i is element number
5. Prepare initial input data Xij where i is element number and j is variable type
6. State run time t_r.

A typical control table is shown in Figure 8.4.2.1

Expert Mechanism (Optimiser):

1. Create an object linking of Witness with VB
2. Launch Witness
3. Load Witness model
4. Import input data Xij from control table to model
5. Run simulation to run time t_r
6. Export simulation output Yik to Excel (an example is shown in Figure 8.4.2.2) Where k is the output type.
7. Manipulate Yik using VB in Excel
8. Assess performance using fn (Objective function)
9. While justification is not breached more than twice in succession, and
10. While other constraints are not breached,

11. Identify the main system constraint element(s) Ei
12. If element Ei is a Machine THEN

    If decreasing cycle time ci is possible AND decreasing ci < limit THEN

        Decrease ci by d_i (decreasing percentage, 10% in this case study)

        Else

        If new element possible THEN

            Add new element and reset ci

        Else

            Do nothing

  If element is Conveyor THEN

      If decreasing index time Ii is possible THEN

          Decrease Ii by d_i (decreasing percentage, 10% in this case study)

      Else

          If new element possible THEN

              Add new element

          Else

              Do nothing

  If element is Buffer THEN

      Increase buffer size by 2

13. Endif
14. Effect changes in Xij
15. End While
16. If steps 9 and 10 TRUE, Go to 4
17. End procedure and document.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1										Run Opt_Mech		IniConsTest		Reset		
2	WITNESS folder		F:\program files\witness 2004													
3	WITNESS model name		F:\Habtom's files\Research\Experimental models\Acme2.mod													
4	Model run time	4000.0														
5				Total runs	0											
6	RESULTS		Run No.													
7			Bars shipped													
8			Cost													
9			Benefit (increase in parts		0	0	0	0	0	0	0	0	0	0	0	
10	**Model Data:**			Orig_data	Orig_qty	No_Change	Mod_qty									
11	CycT_or_Cap (1)	10	Stock	10	1		1									
12	CycT_or_Cap (2)	6	Saw_mach	6	1		1									
13	CycT_or_Cap (3)		CycT_or_Cap													
14	CycT_or_Cap (4)	0.25	Belt_conveyor	0.25	1		1									
15	CycT_or_Cap (5)	60	Coater	60	1		1									
16	CycT_or_Cap (6)	20	Staging_area	20	1		1									
17	CycT_or_Cap (7)	20	Rework	20	1		1									
18	CycT_or_Cap (8)	20	Inspector	20	1		1									
19	CycT_or_Cap (9)	6	Hardner	6	1		1									
20	CycT_or_Cap (10)	7	Loader	7	1		1									
21	CycT_or_Cap (11)	40	Grinder	40	2		2									
22	CycT_or_Cap (12)	4	Unloader	4	1		1									
23	CycT_or_Cap (13)	0.2	Fixture_ret	0.2	1		1									
24	CycT_or_Cap (14)	10	Fixture_store	10	1		1									
25	CycT_or_Cap (15)	10	Valve_stock	10	2		2									
26	CycT_or_Cap (16)	15	Cleaner	15	1		1									
27																
28	Notes:															
29	Machine states							Model Status								
30	0	Off-shift		6	Waiting for labour to cycle				1	Running						
31	1	Idle, waiting parts		7	Waiting for labur to set-up				2	Stopped, waiting for input						
32	2	Busy		8	Waiting for labur to repair				3	Other state(e.g, error message)						
33	3	Blocked							4	Stopped b/c of witness error						
34	4	Setting up														
35	5	Down and repair														

Figure 8.4.2.1 A typical data control table

In summary, the simulator uses data displayed in Excel but controlled by VB, runs the model and generates output. The expert mechanism receives the relevant output data from the simulator, manipulates the data, assesses model performance and generates recommended changes for the next run. The iteration goes on until a limiting factor is reached, at which point the result would be output for documentation.

	A	B	C	D	E	F	G	H	I	J
1										
2	**Final parts**	**464**								
3	**AV.WIP**	**61**								
4										
5	**Imorted Data**									
6	**Sequence No.**	**Element Na**	**Qty**	**Utilisation**	**Setup**	**Down Repair**	**Blocked**	**Idle**	**Util + Set-up**	**Element type**
7	1	Stock	1							4
8	2	Saw_mach	1	26.879	3.457	25.394	44.270	0.000	30.336	2
9	3	CycT_or_Ca	18							11
10	4	Belt_Convey	1	23.554	0.000	0.000	57.218	19.229	23.554	3
11	5	Coater	2	68.100	0.714	0.000	16.160	15.025	68.814	2
12	6	Staging_area	1							4
13	7	Rework	1							4
14	8	Inspector	3	89.702	0.000	0.000	10.298	0.000	89.702	2
15	9	Hardner	1	3.257	0.000	0.000	11.835	0.000	88.165	3
16	10	Loader	1	23.183	0.000	0.000	11.664	65.153	23.183	2
17	11	Grinder	2	65.984	0.000	21.608	0.145	12.263	65.984	2
18	12	Unloader	1	13.257	0.000	0.000	0.000	86.743	13.257	2
19	13	fixture_ret	1	19.543	0.000	0.000	0.000	80.457	19.543	3
20	14	Fixture_store	1							4
21	15	Valve_stock	2							4
22	16	RStream	1							11
23	17	Cleaner	1	0.000	0.571	10.453	0.000	88.975	0.571	2
24	18	Bar	0							1
25	19	Fixture	0							1
26										
27										
28	**Sorted Data**									
29	**Sequence No.**	**Element Na**	**Qty**	**Utilisation**	**Setup**	**Down Repair**	**Blocked**	**Idle**	**Util & Set-up**	**Element type**
30	8	Inspector	3	89.702	0.000	0.000	10.298	0.000	89.702	2
31	5	Coater	2	68.100	0.714	0.000	16.160	15.025	68.814	2
32	11	Grinder	2	65.984	0.000	21.608	0.145	12.263	65.984	2
33	2	Saw_mach	1	26.879	3.457	25.394	44.270	0.000	30.336	2
34	4	Belt_Convey	1	23.554	0.000	0.000	57.218	19.229	23.554	3
35	10	Loader	1	23.183	0.000	0.000	11.664	65.153	23.183	2
36	13	fixture_ret	1	19.543	0.000	0.000	0.000	80.457	19.543	3
37	12	Unloader	1	13.257	0.000	0.000	0.000	86.743	13.257	2
38	9	Hardner	1	3.257	84.908	0.000	11.835	0.000	88.165	3
39	17	Cleaner	1	0.000	0.571	10.453	0.000	88.975	0.571	2
40	1	Stock	1							4
41	3	CycT_or_Ca	18							11
42	6	Staging_area	1							4
43	7	Rework	1							4
44	14	Fixture_store	1							4
45	15	Valve_stock	2							4
46	16	RStream	1							11
47	18	Bar	0							1
48	19	Fixture	0							1

Figure 8.4.2.2. Typical VB manipulated simulation output Yij

8.4.3 Results and Conclusion

Eleven simulation runs were conducted with the summary of results shown in Figure 8.4.3.2. A typical cost-benefit table for a run is also shown in Figure 8.4.3.1. The results indicate that most of the runs could be justified for their respective investments. Models 8 and 10 showed the better net savings, with model 10 significantly better both in savings and its throughput in view of rectifying the current problems of the company.

The reasons for runs 4 and 7 showing negative net savings could be explained in terms of the discrete changes conducted in each iteration. In some instances, the available possible investment on an element could be such that a shift of bottleneck to another element results in the investment not having a significant impact. In case of run 4, for instance, a reduction of cycle time by 10% for element "Inspector" in the preceding run (run 3) costed 24 equivalent parts. In run 4, the bottleneck shifted to element "Coater" resulting in an increase in throughput of only 12 parts. Additionally, in the preceding run, the saving was small such that it did not contribute significantly to the current shift of bottleneck.

The case study model is limited in many respects but has highlighted the basic concept of integrating an optimising element to a manufacturing simulator for automatic results analysis and performance enhancement. The scope of the case study was limited in that there was not a problem of product mix. Thus the constraint income aspect of the TOC was not applied. Additionally, the operating cost was given in general terms as the additional costs of introducing new changes without breaking it into fixed and variable costs. The effect of work in progress was also assumed to be insignificant.

Various performance assessment methods will be incorporated into ensuing case studies to make the environment more versatile for a wider spectrum of simulation scenario. The proposed concept has been designed to handle a mix of different production objectives whereby target figures can be set with each objective, and the system would iterate until those targets are met within specified constraints.

The case study verified and to an extent validated the proposal of integrating a manufacturing simulator with an expert mechanism to enhance performance. It has proved that the integrated system worked as expected. It also substantiated the use of a cusum chart to check data consistency for better optimisation process.

It is worth noting that the case study is an exemplar and uses mainly the TOC as a proven performance enhancing method to illustrate the research concept.

Run No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Valves, x	136	145	144	144	148	148	140	148	144	145	150	152	156	154	140	148	148	152	154	148	148	144	152	147	140
x-T	-11	-2	-3	-3	1	1	-7	1	-3	-2	3	5	9	7	-7	1	1	5	7	1	1	-3	5	0	-7
CuSum	-11	-13	-16	-19	-18	-17	-24	-23	-26	-28	-25	-20	-11	-4	-11	-10	-9	-4	3	4	5	2	7	7	0
Upper mask															41.86	39.4	36.93	34.47	32.01	29.55	27.08	23.14	17.73	11.33	4.924
Lower mask															-41.9	-39.4	-36.9	-34.5	-32	-29.5	-27.1	-23.1	-17.7	-11.3	-4.9

Target, T	147
StdDev	4.924

Ctrl Factors	8.5	8.0	7.5	7.0	6.5	6.0	5.5	4.7	3.6	2.3	1.0

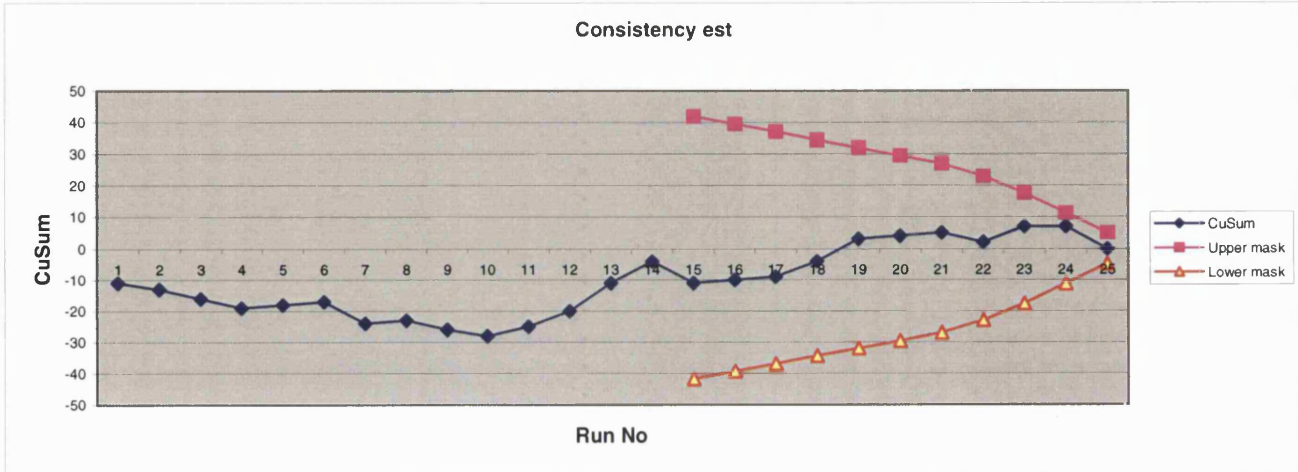


Figure 8.3. Consistency Test (Cusum Test) : Initial model

	Original Data			Costing (equivalent parts)			Modified Data						
Element Name	CycT_or_Cap	Setup_T (min)	Qty	Add New elem	Decrease Cyc_T 10%	Decrease Setup 10%	Mod_cycT (min)	No_CycT Changes	Mod_setupT (min)	No_Set Changes	Mod_Qty	Cost (equiv_part	Cost (£)
Stock	10		1		0.2		10	0			1	0	0
Saw_mach	6	15	1	100	10	20	6	0	15		1	0	0
CycT_or_Cap													0
Belt_conveyor	0.25		1		10		0.25	0			1	0	0
Coater	60	5	1	72	24		48.6	2	5		2	120	30000
Staging_area	20		1				20	0			1	0	0
Rework	20		1		0.2		20	0			1	0	0
Inspector	20	1	1	80	24		14.58	3	1		3	208	52000
Hardner	6		1		24		6	0			1	0	0
Loader	7		1	64			7	0			1	0	0
Grinder	40		2	200	40		40	0			2	0	0
Unloader	4		1	64			4	0			1	0	0
Fixture_ret	0.2		1		0.2		0.2	0			1	0	0
Fixture_store	10		1				10	0			1	0	0
Valve_stock	10		2				10	0			2	0	0
Cleaner	15	50	1	40			15	0	50		1	0	0
										Total cost (Equiv_parts		328	82000
										Final parts		464	116000
										Initial throughput		136	34000

Figure 8.4.3.1 A cost-benefit table for run 11.

Model No.	1	2	3	4	5	6	7	8	9	10	11
Valves shipped	136	168	188	196	260	296	312	392	408	452	464
Cost (in Equiv. parts)	0	24	48	72	104	152	200	224	256	280	328
Benefit (increase in parts)	0	32	52	60	124	160	176	256	272	316	328
Profit (Saving)	0	8	4	-12	20	8	-24	32	16	36	0

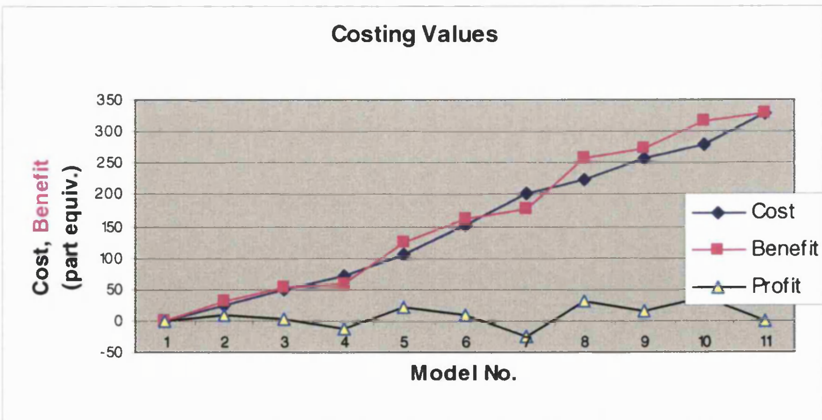


Figure 8.4.3.2 Simulation results and costing

CHAPTER 9

CASE STUDY TWO

(Part of the results of this case study has been published in the proceedings of the Simulation Study Group Workshop, 20-21 March 2002, The Operations Research Society, pp 145-150.)

This chapter presents another case study whose objective showed some addition to the first case study. It demonstrates an application of the expert mechanism to the case study and discusses the results that substantiate the effectiveness and robustness of the expert mechanism.

9.1 Objective of Case Study

Possibilities of improving throughput, reducing operating costs, etc of an existing system can be observed from the analysis of simulation outputs. When the resources are limited, one of the common ways of meeting extra demand is through overtime work. The case study uses the expert mechanism to improve throughput and operating cost within the constraints of existing hardware resources. The improvement is to be done without incurring cost on hardware resources and labour. Here, within the existing resources, attention is given to the constraint elements whose input variable changes - mainly overtime work- can result in improved throughput and cost per unit.

9.2 The Base Model

The model is a flywheel and starter ring gear manufacturing line that was re-planned and extra machines installed to increase capacity. The objective was to increase throughput from 1250 to close to 4000 flywheels per week based on two shift working (4680 minutes per week). It was accepted that some seasonality in demand would occur and overtime would be worked to accommodate these variations. The main changes to the line were: a pair of new rough turning machines, a replacement heat treat and assembly machine, a new fine turning machine and an extra gear hobbing machine on the starter ring line. The diesel flywheel was almost identical to the petrol except that it was approximately 1kg heavier and needed a different starter ring gear that was bought in finish machined. The manufacturing times for the diesel flywheel were identical to the petrol, although there was a change over time required on the rough turning (Table 9.2). The castings for both flywheels and the forging for the ring gear for the petrol were bought in. With the new machines introduced, the system produced around 2280 flywheels per week in total.

The line was initially planned at the time using a flow chart, using work standard and maintenance records to establish the capacity. The tool change details are shown in table 9.2 and a simplified version of the flow chart is shown in Figure 9.2A. The Witness model for the manufacturing line is shown in Figure 9.2B.

Table 9.2. Tool change details (All times in minutes)

Tool	*Operations*	*Change time (normal distribution; mean, σ)*	*Changed by*
Rough turn op 10 (new)			
1	100	2,0.1	Operator
2	100	2,0.1	Operator
3	200	3,0.2	Operator
4	500	3,0.2	Operator
5	500	5,0.3	Operator
Rough turn op 20 (new)			
1	100	2,0.1	Operator
2	200	3,0.2	Operator
3	200	3,0.2	Operator
4	300	5,0.3	Operator
5	500	5,0.4	Operator
6	700	10,0.5	Operator
Rough turn Ops 10 & 20 (old)			
1	80	3,0.2	Operator
2	150	5,0.3	Operator

3	200	5,0.4	Operator
4	250	3,0.4	Operator
5	300	3,0.4	Operator
6	350	5,0.4	Operator
7	400	5,0.5	Operator
8	600	5.0.4	Operator
Change over from petrol to and from diesel 20 by operator; Load and unload 0.2 by operator			
Fine bore op 30			
1	1000	3,0.3	Operator
Drill and tap op 40			
1	500	10,0.5	Operator
2	1000	30,1	Operator
3	1500	15,1	Operator
Load and unload 0.2 by operator			
Fine turn op 60			
1	250	2,0.1	Operator
2	500	3,0.2	Operator
Balance op 70			
1	1000	4,0.3	Operator
Bore & turn op 20			
1	500	5,0.2	Operator
Load & unload 0.2 by operator			
Hob op 30			
1	5000	10	Operator
Load & unload 1.5 per 18 batch by operator			
Chamfer op 40			
1	500	10,2	Operator

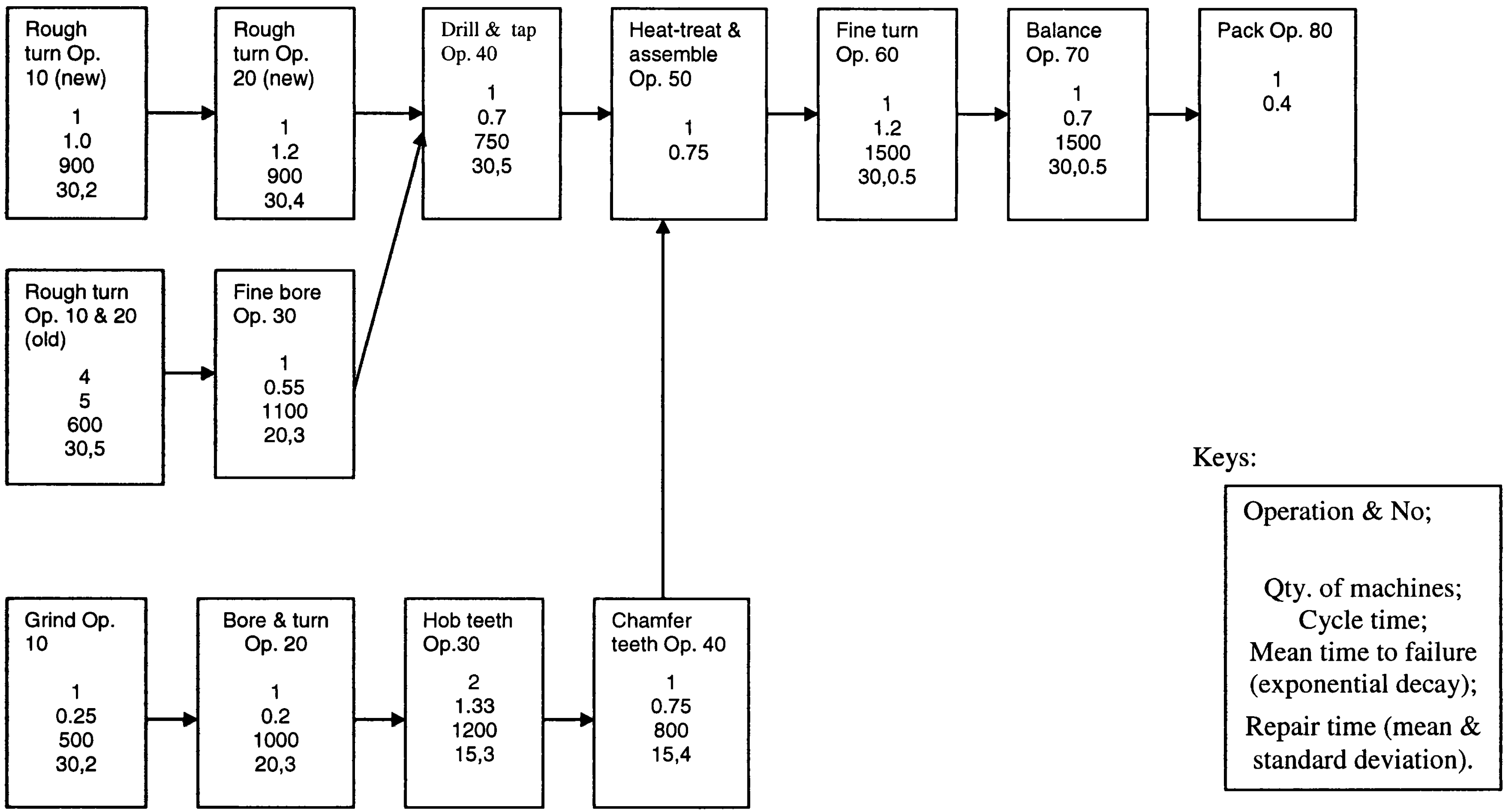


Figure 9.2A Flow chart for modelling the Flywheel & Starter ring manufacture

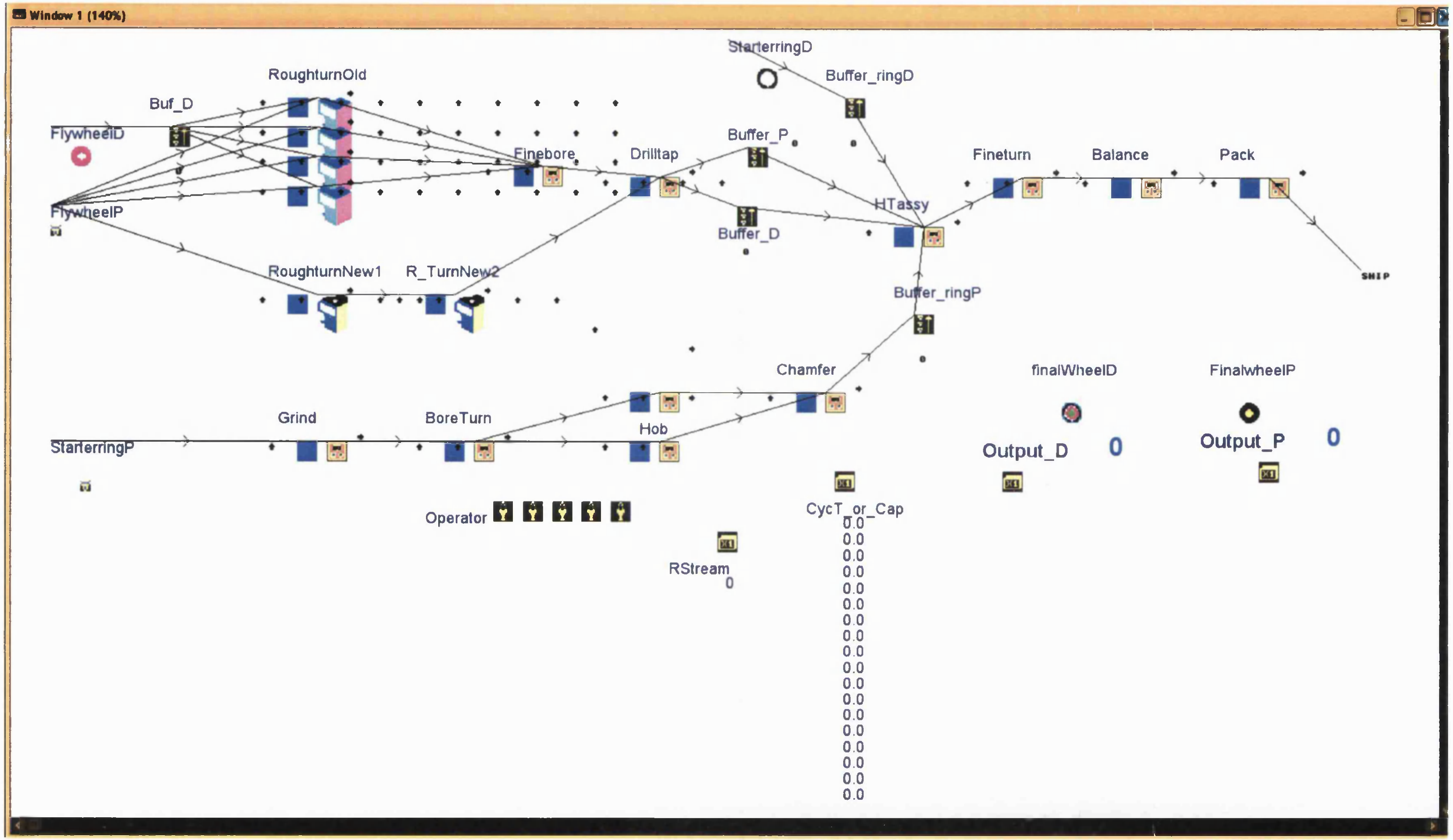


Figure 9.2B. Model of the Flywheel manufacturing line

9.3 Model Validation and Consistency Check

To ensure that the model output variability is within an acceptable level for effective performance optimisation, a cusum chart was used. As stated, a cusum chart is recognized as the best statistical process control (SPC) chart for detecting small changes that would not be detected by a Shewhart chart (Owen, 1989). This ability of the cusum chart gives the benefit of long-term development potential to monitor and control the simulation. The cusum chart would enable the simulation to automatically detect whether the model output data is under statistical control.

As described in previous chapters and for completeness restated here, the cusum chart requires two pieces of information to initiate the system; they are the target value and the sample standard deviation. It is suggested that the mean of the initial results is the target value and the sample standard deviation s is the value used to calculate the mask limits. As the standard a minimum of 25 readings are required to calculate these variables.

A cusum chart in essence is about slope: if the cusum points are approximately horizontal then the readings are near the target value and the process is under control. If the slope is upward then the readings are above target and the process may be out of control. If the readings are less than target, the slope will be downward.

A mask is used to determine if the slope is too great for the process to be considered under control. The focus of the mask is placed on the last reading; all the previous readings should fall within the arms of the masks. If they do, then the process is under control.

The semi-parabolic mask that was recommended is equivalent to a 2.65σ action limit and 1.65σ warning limit. Other mask designs are available; see BS 5703 (2003) for further details.

Table 9.3A shows the results for the first trial (experiment No. 1). The value of s used to draw the mask is the sample standard deviation (Coleman et al, 1996) of the x-T values from the initial results. This value can be used, as the standard deviation of the results is not expected to change during the run. When the optimization process is completed it will also recheck the model to ensure that it still maintains data accuracy.

The existing system, with the relevant operating data shown in table 1, produces around 2280 flywheels (~2180 flywheelP type and ~100 flywheelD type). The simulation model for the existing system was built with randomness as experienced by the real system; the model output was then checked against the existing data and showed good correlation.

The data consistency of the model was then checked by conducting 25 runs changing all the random number streams for each run. This check for consistency is a part of sensitivity analysis as recommended by Balci (1994).

Outputs from 25 runs were recorded and a Cusum (90% confidence) was conducted as shown in Figure 9.3A. The mask used is a C2 semi-parabolic mask as defined in BS 5703 part 3 (2003). It could be seen from the graph that the data were all between the upper mask and the lower mask, indicating good consistency.

A similar consistency check was conducted on the last model (Figure 9.3B) and it again showed satisfactory stability.

The purpose of the stability check is to ensure that the model output data is under control in terms of variability before the optimisation process can commence. If it does not then there is no guarantee that optimisation process could proceed effectively.

Run No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
FlywheelP	2149	2232	2239	2095	2101	2225	2067	2162	2209	2176	2188	2199	2220	2156	2256	2185	2169	2151	2212	2279	2265	2167	2178	2179	2163
FlywheelD	99	99	97	99	97	98	99	99	98	100	98	98	98	99	99	99	99	99	98	100	99	99	99	98	99
Total parts,x	2248	2331	2336	2194	2198	2323	2166	2261	2307	2276	2286	2297	2318	2255	2355	2284	2268	2250	2310	2379	2364	2266	2277	2277	2262
x-T	-35.5	47.5	52.5	-89.5	-85.5	39.5	-117.5	-22.5	23.5	-7.5	2.5	13.5	34.5	-28.5	71.5	0.5	-15.5	-33.5	26.5	95.5	80.5	-17.5	-6.5	-6.5	-21.5
CuSum	-35.5	12.0	64.4	-25.1	-110.6	-71.1	-188.6	-211.2	-187.7	-195.2	-192.7	-179.2	-144.8	-173.3	-101.8	-101.3	-116.8	-150.4	-123.9	-28.4	52.1	34.6	28.0	21.5	0.0
Upper mask															438.6	412.8	387.0	361.2	335.4	309.6	283.8	242.5	185.8	118.7	51.6
Lower mask															-438.6	-412.8	-387.0	-361.2	-335.4	-309.6	-283.8	-242.5	-185.8	-118.7	-51.6

Target, T	2284
StdDev	51.6

Ctrl Factors	8.5	8.0	7.5	7.0	6.5	6.0	5.5	4.7	3.6	2.3	1.0

Table 9.3A Results for Consistency test (Cusum Test) - Initial model

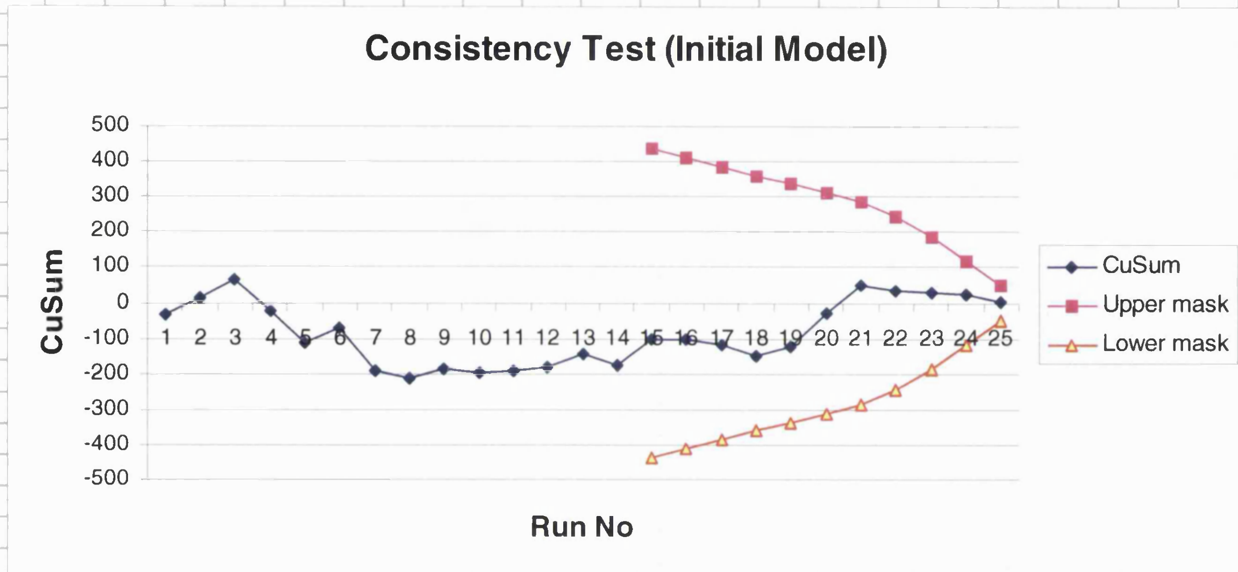


Figure 9.3A Consistency test on Initial Model

Run No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
FlywheelP	3758	3875	3861	3758	3930	3898	3724	3627	3909	3843	3929	3802	3972	3839	3838	4001	4006	3911	3834	3988	3987	3691	3845	3762
FlywheelD	95	95	100	100	100	96	95	100	100	95	99	100	100	100	99	100	100	95	100	95	100	96	95	100
Total parts,x	3853	3970	3961	3858	4030	3994	3819	3727	4009	3938	4028	3902	4072	3939	3937	4101	4106	4006	3934	4083	4087	3787	3940	3862
x-T	-100.2	16.8	7.8	-95.2	76.8	40.8	-134.2	-226.2	55.8	-15.2	74.8	-51.2	118.8	-14.2	-16.2	147.8	152.8	52.8	-19.2	129.8	133.8	-166.2	-13.2	-91.2
CuSum	-100.2	-83.5	-75.7	-171.0	-94.2	-53.4	-187.7	-413.9	-358.2	-373.4	-298.6	-349.9	-231.1	-245.4	-261.6	-113.8	38.9	91.7	72.4	202.2	336.0	169.7	156.5	65.2
Upper mask															860.2	809.6	759.0	708.4	657.8	607.2	556.6	475.6	364.3	232.8
Lower mask															-860.2	-809.6	-759.0	-708.4	-657.8	-607.2	-556.6	-475.6	-364.3	-232.8

Target, T	3953.2
StdDev	101.2

Ctrl Factors	8.5	8.0	7.5	7.0	6.5	6.0	5.5	4.7	3.6	2.3

Table 9.3B Results for consistency test (Cusum Test) - Last model

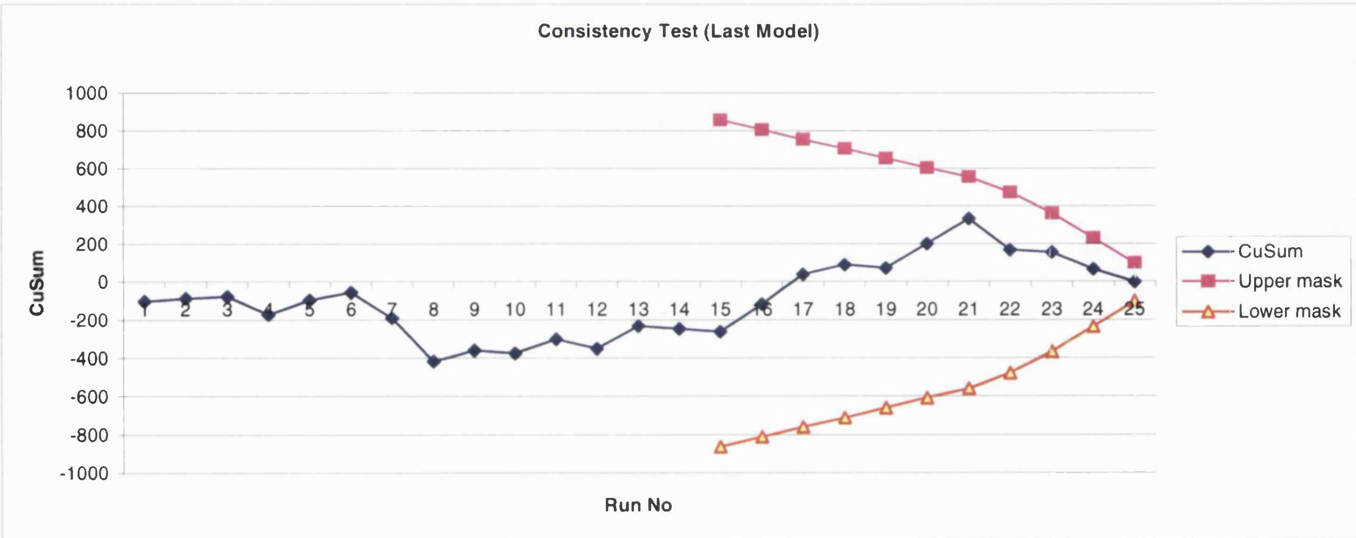


Figure 9.3B Consistency test on Last Model

9.4 Optimisation

For automatic performance enhancement on the simulation system, bypass balancing backed by Theory of Constraints within the existing shop floor system was used. Bypass balancing improves throughput and unit operating costs without requiring extra hard resources to balance the system. Only those resources acting as bottlenecks at the time of the run would be allowed to work overtime. For simplicity, in every adjusted run, the system increments an over-time work of 2 hours at the bottleneck workstation. As the company recommends work up to a maximum of two shifts, the maximum allowable overtime work hours is 40 (one additional shift).

In other words, the optimising mechanism which is written in Visual Basic receives the relevant simulation results from Witness, transfers the data to Excel, arranges the data; then it assesses their values for performance and takes adjusting actions on the input data for the next run. The summary of the optimizing steps are as follows:

Data Preparation (Control Table):

1. Reset control table
2. State the model file to launch (open)
3. State values for constraints
4. Download list of model elements Ei (Optional), where i is element number
5. Prepare initial input data Xij where i is element number and j is variable type
6. State run time t_r.

Expert Mechanism (Optimiser):

1. Create an object linking of Witness with VB
2. Launch Witness
3. Load Witness model
4. Import input data X_{ij} from control table to model
5. Run simulation to run time t_r
6. Export simulation output Y_{ik} to Excel (an example is shown in Figure 8.3.2.2)

Where k is the output type.

7. Manipulate Y_{ik} using VB in Excel
8. Assess performance using F_n (Objective function)
9. While justification is not breached, and
10. While other constraints are not breached,
11. Identify the main system constraint element(s) E_i

    If work-hour < limit THEN

        Increment No of changes to the element by 1

        Increase work-hour by 2

        Reduce process time p_i proportionally

        Calculate the running cost using equation 9.4

    Else

        Give message "Maximum possible overtime reached!"

        Go to step16

12. Endif
13. Effect changes in X_{ij}
14. End While
15. If steps 9 and 10 TRUE, Go to 4
16. End procedure and document.

The procedure that describes the balancing action for workstations in Visual Basic is shown in Appendix B.

The throughput, modified work hours and calculated costs for each run are shown in table 9.4. It is assumed that the system is in steady state, the effect of work in progress due to over-time work is negligible and the variable overhead costs are proportional to the work hours.

The operating cost (Cop) was calculated using the following formula

$$Cop = C_f + (C_{ov} + C_{la}) * (W_h/W_{ho}) + C_{ma} * X \qquad (9.1)$$

Where

C_f – Fixed cost per week

C_{ov} – Variable overhead cost per week

C_{la} – Labour cost per week

W_h – Total Work hour

W_{ho}- Initial total work-hour

C_{ma} – Material cost per unit

X – Total parts made

The modified costs included a 50% mark up of overtime labour rate as was customary in the company. This modified cost was calculated as

$$C_{op\text{-}mod} = C_f + (C_{ov} + C_{la})*(W_h/W_{ho}) + C_{ma} * X + (0.5*C_{la}*(W_h - W_{ho})/W_{ho} \qquad (9.2)$$

The adjustment to the process time p_i of the constraint station is made using the original process time p_{io} as

$$p_i = (W_{ho} / W_h) * p_{io} \qquad (9.3)$$

9.5 Results and Conclusion

The maximum overtime work hour was bridged after the twenty first run. The twenty one simulation-optimisation runs conducted on the existing resource are shown in Table 9.4 and Figure 9.4. At the end of runs where one workstation had to work to an over-time of 40 hours per week, the total throughput increased by 70 percent and the cost per flywheel was reduced by 21 percent. There was an average increase of 17 flywheels in every 2-hour of over-time work as compared to under 9 flywheels in the existing system.

As there often is a fluctuation in demand, the table (and the graph) was intended to be used for selecting the near best combination that would fit the production schedule at hand.

The case study has demonstrated the basic concept that the expert mechanism has properly controlled the simulator and with its embedded knowledge base, managed to conduct an automatic optimisation of a simulation process in manufacturing – in this case, where the resources are limited, by controlling the overtime work. It also demonstrated that the model could be checked for statistical data consistency automatically.

The experiment proved that the integrated system worked as intended and the results substantiated the effectiveness of using an expert mechanism to assist in interpreting manufacturing simulation output data, assessing performance and in enhancing performance. The rules that apply to overtime work allocation and the calculation of operating cost are among the main additions to the body of knowledge in this case study.

It should be understood that the effect of the overtime work of constraint workstations in the manufacturing line was introduced by proportionally adjusting the processing times of the workstations. This leads to suppressing the effects of work in progress. In this special case, the effect of WIP was negligible but for other similar work where WIP is important, the overtime has to be included by adjusting the individual shifts. The model will of course be more complex as it will require separate shift data for each workstation.

Total runs	21																					
Run No.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Parts made, X		2248	2334	2427	2484	2538	2607	2743	2822	2913	3005	3098	3206	3294	3291	3396	3423	3549	3667	3713	3792	3824
	Modified Data					Work hours																
RoughturnOld	5.00	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
RoughturnNew	0.48	40	40	42	44	46	48	50	52	54	56	58	60	60	62	64	64	66	68	70	70	70
R_turnNew2	0.45	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	76	78
Finebore	0.55	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Drilltap	0.25	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
Hassy	0.48	40	40	40	40	42	42	44	46	48	48	50	52	52	52	54	54	56	56	58	58	58
Grind	0.25	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
BoreTurn	1.20	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Hob	16.24	40	40	40	40	40	40	42	42	44	46	46	48	50	50	50	50	52	54	54	56	56
Chamfer	0.46	40	40	40	40	42	42	44	46	48	48	50	52	52	52	54	54	56	58	58	58	60
Fineturn	0.94	40	40	40	40	40	40	40	40	40	40	42	42	44	44	44	46	46	48	48	50	50
Balance	0.47	40	40	40	40	40	40	42	44	44	46	48	48	50	50	52	52	52	54	54	56	56
Pack	0.40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Total work_hour/week		520	524	530	536	546	552	566	578	590	600	614	626	636	642	654	660	672	686	694	702	708
Op_cost-flat (£)		21144	21444	21785	22018	22284	22553	23107	23468	23866	24246	24670	25119	25487	25540	25980	26123	26625	27125	27346	27666	27824
Cost/part (£)		9.41	9.19	8.98	8.86	8.78	8.65	8.42	8.32	8.19	8.07	7.96	7.83	7.74	7.76	7.65	7.63	7.50	7.40	7.36	7.30	7.28
Op_cost-mod (£)		21144	21450	21802	22046	22329	22609	23186	23569	23987	24384	24833	25302	25687	25751	26211	26365	26889	27412	27647	27981	28150
Cost/part (£)		9.41	9.19	8.98	8.88	8.80	8.67	8.45	8.35	8.23	8.11	8.02	7.89	7.80	7.82	7.72	7.70	7.58	7.48	7.45	7.38	7.36

Fixed cost/week, Cf (£)	9000
Overhead cost/week, Cov (£)	3600
Labour cost/week, Cla (£)	1800
Material cost/part, Cma (£)	3

Operating Cost(flat), Cop = Cf + (Cov + Cla)*(Wh/Who) + Cma * X

Operating Cost(mod), Cop = Cf + (Cov + Cla)*(Wh/Who) + Cma * X + (0.5*Cla*(Wh - Who)/Who)

Table 9.4. Throughput, modified data, and operating costs

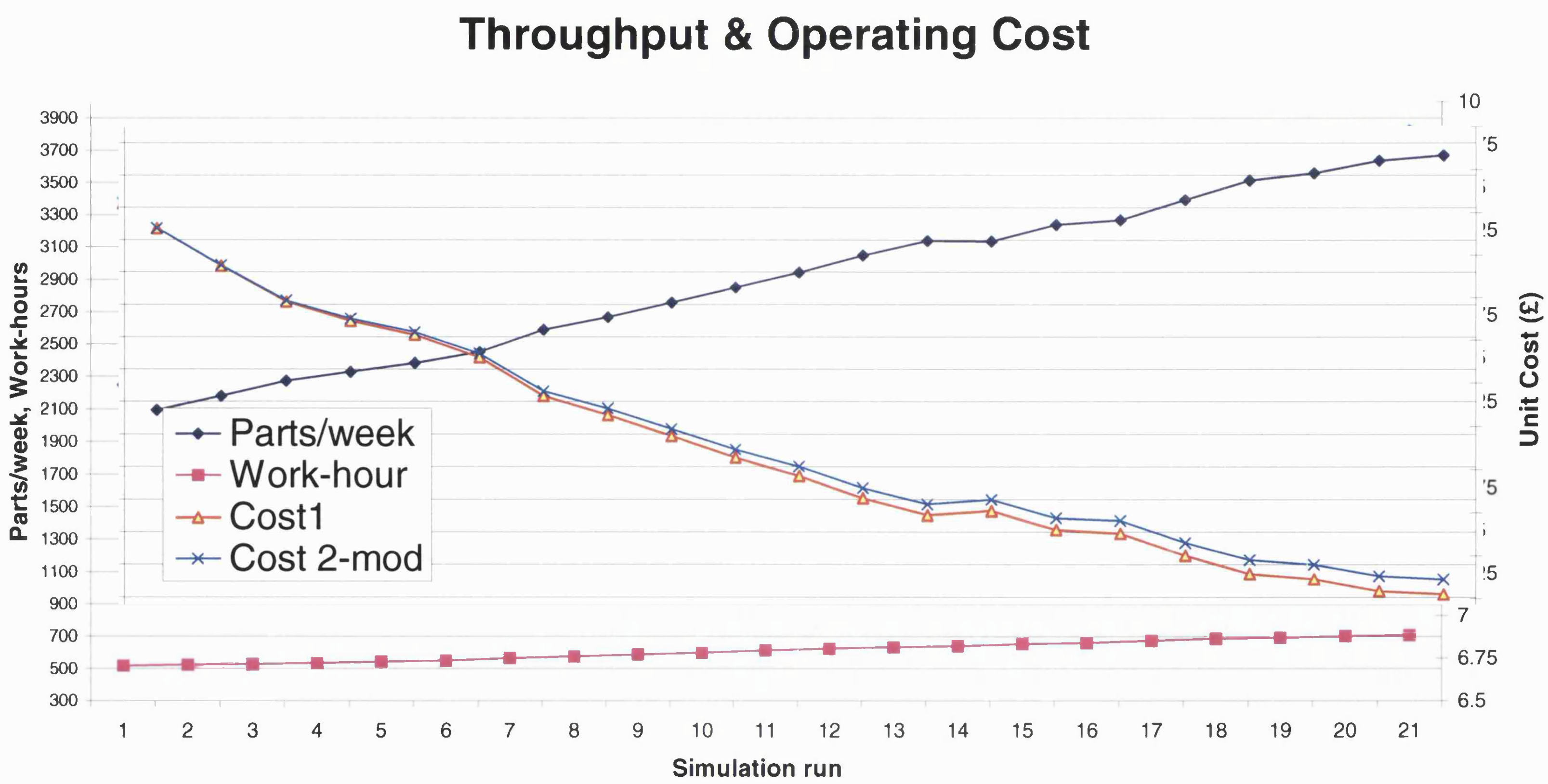


Figure 9.4 Throughput and Unit costs.

CHAPTER 10
CASE STUDY THREE

This chapter covers a further case study from a world class company with real data and a company validated simulation model. Following a satisfactory set of results, it discusses the success of the research and validates the expert mechanism by comparing it with one of the leading commercial optimisation packages.

10.1 Objective of case Study

This case study is based on a real model from a world class company, SELEX Sensors and Airborne Systems (S&AS), Basildon, Essex. Its main objective was to validate the optimisation system of this project, expert mechanism, with real manufacturing data and to compare its strengths and weaknesses with a prominent commercial optimiser. The simulation model has been built by the company and the Expert Mechanism was expected to conduct performance optimisation without requiring a modification to the elements of the simulation model by retrieving input data variables from the model, running the model, gathering output data, assessing performance, identifying the main variable(s) to change consistent with the constraints, effecting the input variable(s) changes, running the model, and repeating the optimisation process until the constraints have limited the iteration or the required performance has been reached.

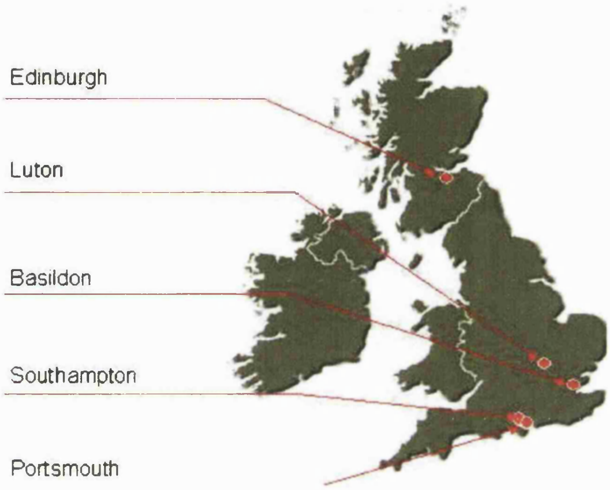
Conducting data consistency check was not needed in this case study because the data, as provided from the company, were predominantly of a deterministic nature; thus significant variability in output data was not expected.

It should be noted that due to product sensitivity and confidentiality issues most of the model descriptions have been de-characterised but the model core (flow and rules) has not been altered.

10.2 Company Background

SELEX Sensors and Airborne Systems (S&AS), part of the Finmeccanica defence electronics sector, employs 7,600 people with operations in Italy, England, Scotland and the United States. Specialising in world-class integrated sensor solutions, SELEX S&AS is a leader in surveillance, protection, tracking, targeting, navigation & control and imaging systems.

In the UK SELEX S&AS is the foremost supplier of electronic systems for military platforms in the air, at sea and on land. It has major sites in Southern England and Scotland employing 4,400 people.



Key Technologies Include:

- Airborne Radar Systems
- Electronic Warfare Systems
- Electro Optical Systems
- Military Lasers

The Basildon site has in excess of 1,000 employees, with 50 years experience in manufacturing state of the art military guidance systems and civil aeronautics for world markets. Current products include:

- Missile seekers
- Cooled and uncooled thermal imaging systems
- Radio frequency
- Civil communications equipment
- Stabilized platforms
- Optical assemblies

The current manufacturing environment consists of a mix of high/low volume manufacturing cells or lines and high/low value products. Production volumes vary from one-offs to 135 units/month depending on the project. The high volume lines have been the traditional application areas of in-house Witness manufacturing simulation and optimization, with the following generic objectives:

- verification of production set-up and output
- determining resources i.e. labour, material, equipment
- production area layout
- production recovery studies
- business continuity planning
- capital investment justification

The modelling process involves the development of a simulation model, with an Excel input/output interface, which is validated against the actual production environment. This model is then used to carry out performance enhancing runs, either using the simulation model and Excel input/output interface or using the add-on Witness Optimizer module. The former allows quick, simple scenarios to be executed by production managers, with no Witness experience requirements. The latter allows more complex optimization scenarios to be carried out utilizing optimization algorithms, mainly simulated annealing and tabu search, but is constrained by the fact that it requires an expert Witness user and a development PC asset. The Witness Optimiser is considered as one of the most successful commercial simulation optimisation packages.

The same validated model was also used by the Expert mechanism to test the capability of the Expert mechanism.

10.3 The Base Model

The modelled system was part of a contract awarded to BAE SYSTEMS Avionics (now SELEX S&AS) at Basildon. The scope of work for this site was to manufacture an Infrared device. The particular project was a significant win in terms of its sales value and the prestige of working with a major US partner.

It was also important with regards to the manufacturing philosophy that needed to be applied. This was formulated due to the high volume requirements for a product that is technically challenging to produce. The programme schedule was aggressive and new ways of manufacturing were required for its success. Equally important was the need to maximise the highest quality due to the nature of this product and also strict controls with regards to build history, failure history and configuration control.

The model was focused on the main assembly cell of the product, which was specifically designed and built as a class 10,000 clean room. The cell is dedicated to the manufacture of the product and it encompasses assembly, inspection and test activities.

The manufacturing environment is a traditional assembly/inspection/test (AIT) process, heavily dependent on labour input, with automated equipment only used to carry out the various test stages.

Batch manufacturing is traditionally used at the Basildon factory due to the volume and complexity of the products. The high volume requirements of the new programme required a re-think of Basildon's traditional manufacturing processes. A flow line production process was designed and developed, being fed from a number of sub-assembly stations. This method allowed the following benefits to be realized:

- Reduced set-up times, due to dedicated workstations with all the tools and materials required to perform the operation. The operators move through the flow line with the product. No set-ups every time a new batch hits the flow line.
- Minimum risk and exposure to any quality problems as the product flowed to test and any issues are immediately identified before a large batch is produced.
- Bottlenecks were minimized as the flow line was designed for balance with regards to workstation target times.
- Handling and 'dead-time' were minimized with each workstation adjacent to each other to increase flow speed.

- Dedicated test station and environmental stations, with no product or facility sharing.
- Electronic data environment, eliminated paper in favour of electronic methods via manufacturing portal.

The cell has been designed as a high-volume production environment with the capacity to handle up to 150 units per month. Current production requirements are fixed on a month-to-month basis and are around 120 units per month.

As mentioned above, the system is a dedicated production cell, so there is no product variety within the process. Some internal variety exists due to reworked units.

A flow chart of the production process is shown in Figure 10.3.1 and in a WITNESS environment in Appendix E.

10.4 Model objectives and constraints

The overall simulation model was mainly developed to support the production set-up, verify the monthly production output and allocate labour resources with their utilizations. It was also intended to be used as a tool to run 'what-if' scenarios and optimization studies to identify improvement opportunities during production or, recovery scenarios in case of material shortages, significant equipment downtime or low output. As such, it was intended as both a tactical and strategic decision-making tool for the production manager and the Operations management.

The Witness model was developed using an Excel template as an input/output user interface that also allowed different scenarios to be evaluated by altering values through the template. Although this method allowed a level of 'what-if' analysis to be carried out, it was not a conclusive optimization process as it was both slow and was only considering a small number of the potential combinations. Furthermore, it was not taking into account any cost vs. benefit considerations. Hence, it was decided to use the built-in Witness Optimizer (version 4.2) module to carry out the optimization process.

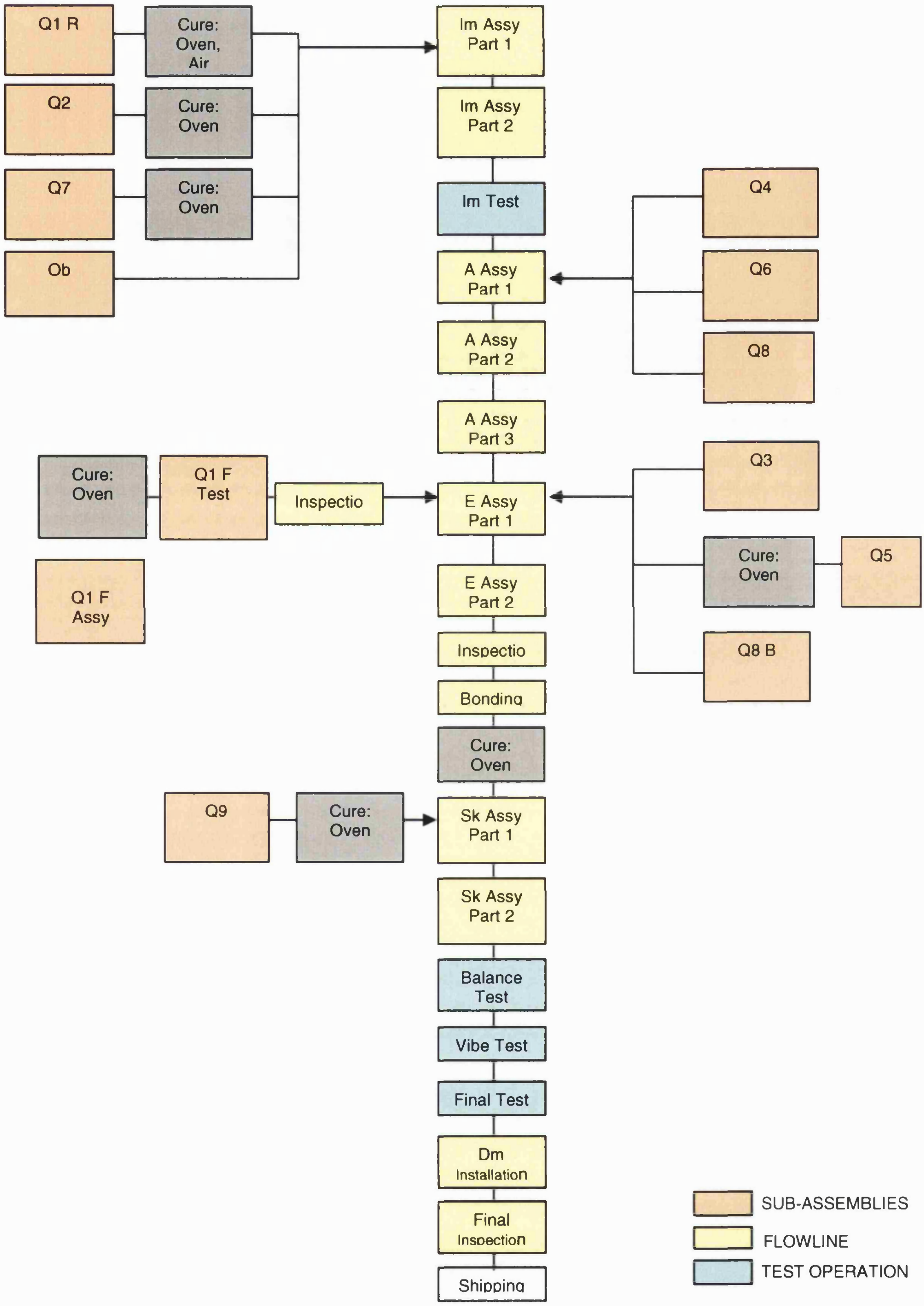


Figure 10.3.1 Production Process Flow (de-characterised for confidentiality)

The main challenge has always been to increase production output with minimum resources and to maximize profit. There is a very fine balance between these three manufacturing considerations, as inappropriate application of resources (material and labour) could increase monthly output but generate a loss to the business due to the cost implications of the applied resources.

The particular optimization model used in this study was based on the latest version of the simulation model and it was developed in order to identify the optimum combination of a set of production parameters to concurrently maximize *profit* and *output*. The optimization variables were the:

- Number of Assembly labour resources
- Number of Test Resources
- Weekly Material Input quantity, and
- Operation cycle times for OP0011, OP0015, OP0031, OP0035, OP0038, OP0051, OP0055, OP0061, OP0071, OP0081 and OP0085.

An optimization constraint was included that limited the combined labour resources of Assembly (standard week shift) and Assembly Weekend (standard week and Saturday shift) to a maximum of 16 people. Additionally, the maximum utilisation of operators is realistically about 85 percent. The cost implication of the first three variables is of prior importance.

Initial Model:

The model of the existing production line was built using Witness 2004, and through multiple simulation trials, the model warm-up time was established at 90,720 minutes i.e. 9 weeks from the model initiation (Figure 10.4.1). The run-time for each evaluation was set at one month's production (43,200 minutes).

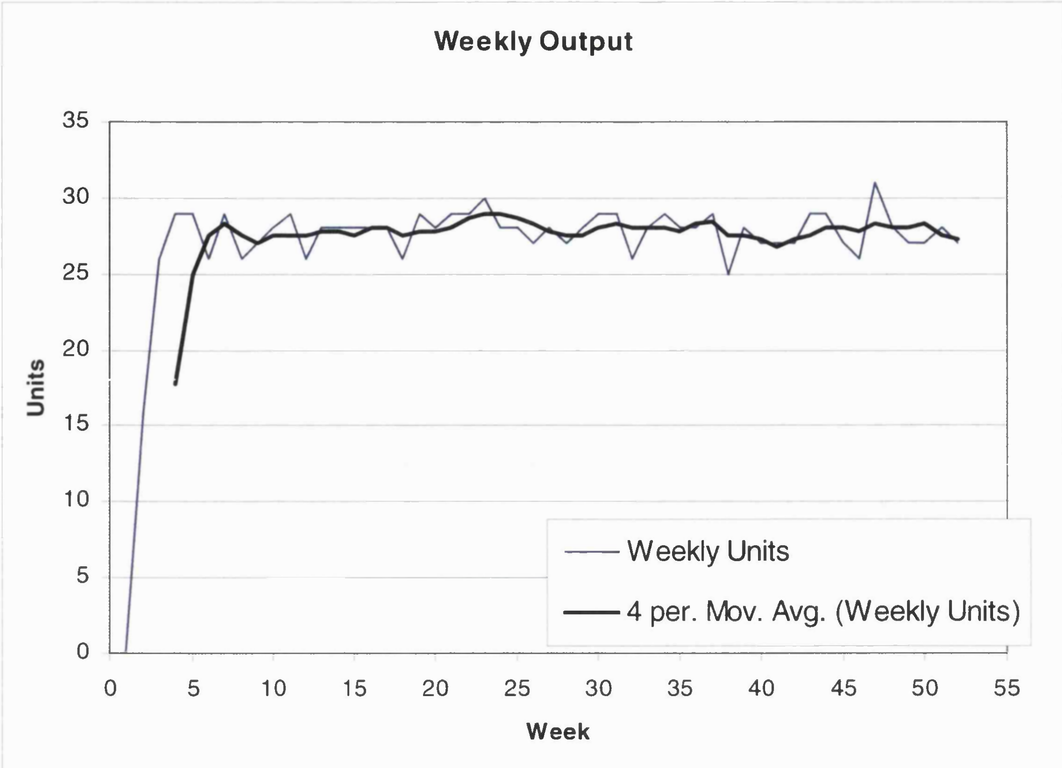


Figure 10.4.1 Establishing model Warm-up period

The initial model was run and compared with the real system. The average output proved to be within 2 percent of the real output which is considered to validate the model.

10.5 Optimisation

Two optimisation systems were used to find the near-optimum value of the objective function (Profit) for the model: Witness Optimizer from Lanner Group and the Expert Mechanism created in this research.

The cost-benefit considerations were set based on normalised weightings to reflect actual considerations as profit units.

Variable	Initial setting	Profit unit
Output (TSk parts)	120	1 per additional part
Weekly Input (Raw parts)	30	-3 per additional part
No. of Assembly operators (Op_Assembly)	6	-1 per additional operator
No. of Assembly Weekend operators (OpAssemblyWkend)	5	-1.5 per additional operator
No. of Test operators (Op_Test)	2	-1 per additional operator

Table 10.4.1 Cost-benefit weightings for optimisation.

10.5.1 Witness Optimizer

The Witness Optimizer uses an Adaptive Thermostatistical Simulated Annealing (TSA) algorithm which is predominantly a simulated annealing algorithm that incorporates elements of tabu search.

The profit function for the Lanner Optimizer was set as:

```
O_Profit = 0.0
IF TSk <= 120              (TSk is the output part)
    O_Income = 0
ELSE
    O_Income = (TSk - 120) * 1
ENDIF
IF WeeklyInput <= 30       (WeeklyInput is the  parts input to the system per week)
    O_MaterialCost = 0
ELSE
    O_MaterialCost = (WeeklyInput - 30) * 3
ENDIF
IF NQTY (Assembly) <= 6
    O_LabourCost1 = 0
ELSE
    O_LabourCost1 = (NQTY (Assembly) - 6) * 1
ENDIF
IF NQTY (AssemblyWeekend) <= 5
    O_LabourCost2 = 0
```

```
ELSE
    O_LabourCost2 = (NQTY (AssemblyWeekend) - 5) * 1.5
ENDIF
IF NQTY (Test) <= 2
    O_LabourCost3 = 0
ELSE
    O_LabourCost3 = (NQTY (Test) - 2) * 1
ENDIF
O_LabourCost = O_LabourCost1 + O_LabourCost2 + O_LabourCost3
O_Cost = O_MaterialCost + O_LabourCost
O_Profit = O_Income - O_Cost
RETURN O_Profit
```

Other settings are shown in Appendix C.

10.4 2 Expert Mechanism

The expert mechanism, taking the costs and constraints into account, used the Theory of Constraints as the main method of identifying the bottleneck(s), manipulated the simulation outputs and effected changes to the most influencing input variables. The primary steps are similar to the previous two case studies. *The main addition to the knowledge base is that once the near-optimised setting has been obtained, an additional algorithm runs to reduce the cost on the setting by reducing underutilised resources. This was achieved by looking at the fraction of unavoidable time* U_i *(busy time + setup time) of the resource (in this case study, operator) and the quantity of the resource element in the model* $NQTY(E_i)$. The process is as follows:

While external constraints are not broken,

If $NQTY(E_i) * (1-U_i) > 1$ then Set $NQTY(E_i)= NQTY(E_i) - 1$

The model was then run. The iteration continued until no more reduction was possible.

10.6 Results and Conclusions

The results from the optimisation process of the Witness Optimizer are shown in table 10.5.1 and that of the Expert mechanism in Table10.5.2, and in Figure 10.5.1. Runs

18, 19 and 20 with the Expert mechanism (Table 10.5.2) are results of the cost reduction algorithm on the near optimum value.

Profit unit (O_profit)	19.5	19.5	19.0	19.0	19.0	18.5	18.5	18.5	18.5	18.5
Output (TSk)	147	147	143	143	143	147	147	147	146	147
WeeklyInput	32	32	31	31	31	32	32	32	32	32
Op_Assembly	6	6	7	7	7	7	7	6	7	7
Op_AssemblyWkend	6	6	5	5	6	6	6	6	6	6
Op_Test	2	2	2	2	2	2	2	2	2	2
Run No.	453	358	203	204	238	146	149	383	384	456

Table 10.5.1 Best 10 results from Witness Optimizer (Total runs: 560)

The results show that the Expert mechanism converged to the near optimum solution more efficiently compared to the Witness Optimizer. This could be attributed to the fact that the Lanner Optimizer works on systematic search for the best solution from a population of possible alternatives whereas the Expert Mechanism attacks the area where there is a definite weakness. The Expert mechanism took account of the working constraint of maximum 85 percent utilisation which was actually not considered by the Witness Optimizer. In fact the utilisation of the Assembly operator was 98 percent in the optimum solution of Witness Optimiser. Part of the manipulated simulation output data used by the Expert mechanism is shown in Table 10.5.3. Columns 4 to 9 are percentages of the available shift time.

The time to reach the optimum for the Expert mechanism was a fraction of that of the Witness Optimizer. Although it is difficult to compare the actual optimisation run times, the following can give an indication of how fast the Expert Mechanism was compared to the Witness Optimiser:

Witness Optimiser was run on a Pentium4 computer, 2.8 GHz processor, 512MB RAM, and took over 2 hours to converge to a solution. The Expert mechanism run on a Pentium centrino computer, 1.4GHz, 512 MB RAM, took about 10 minutes.

This real industrial case study has proved that the Expert mechanism has a very high potential to be used as a generic optimisation tool for manufacturing systems.

In general terms, SELEX S & AS has felt that the optimization study has shown, using standard optimization techniques, that a more powerful and conclusive approach can be reached in identifying better production set-ups that would take a considerable amount of time to achieve using the traditional 'what-if' approach. Furthermore, a series of critical parameters can be linked and evaluated concurrently, which has given visibility to the fact that increased unit production does not always imply higher profit. It has also allowed an insight into how production can be tailored towards sales or profit, depending on the prevailing business requirements.

															*			*1	*2	*3
Profit unit	-1.0	8.5	10.0	8.5	8.0	15.5	12.0	9.5	2.0	**19.5**	15.0	12.5	-7.0	15.5	18.0	18.0	15.5	**19.0**	**17.0**	**16.0**
Monthly Output	119	132	135	135	131	142	140	139	128	**147**	146	145	122	148	**152**	152	151	**152**	**150**	**147**
WeeklyInput	30	30	30	30	31	31	31	31	32	**32**	32	32	33	33	**33**	33	33	**33**	**33**	**33**
Op_Assembly	6	7	7	7	6	7	7	7	6	**6**	7	7	6	7	**7**	7	7	**6**	**5**	**4**
Op_AssyWkend	5	6	7	8	5	6	7	8	5	**6**	7	8	5	6	**7**	8	9	**7**	**7**	**7**
Op_Test	2	3	3	3	2	3	3	3	2	**2**	3	3	2	3	**3**	3	3	**3**	**3**	**3**
Run No.	1	2	3	4	5	6	7	8	9	**10**	11	12	13	14	**15**	16	17	**18**	19	20

Table 10.5.2 Optimisation results using the expert mechanism (SELEX S & AS model)

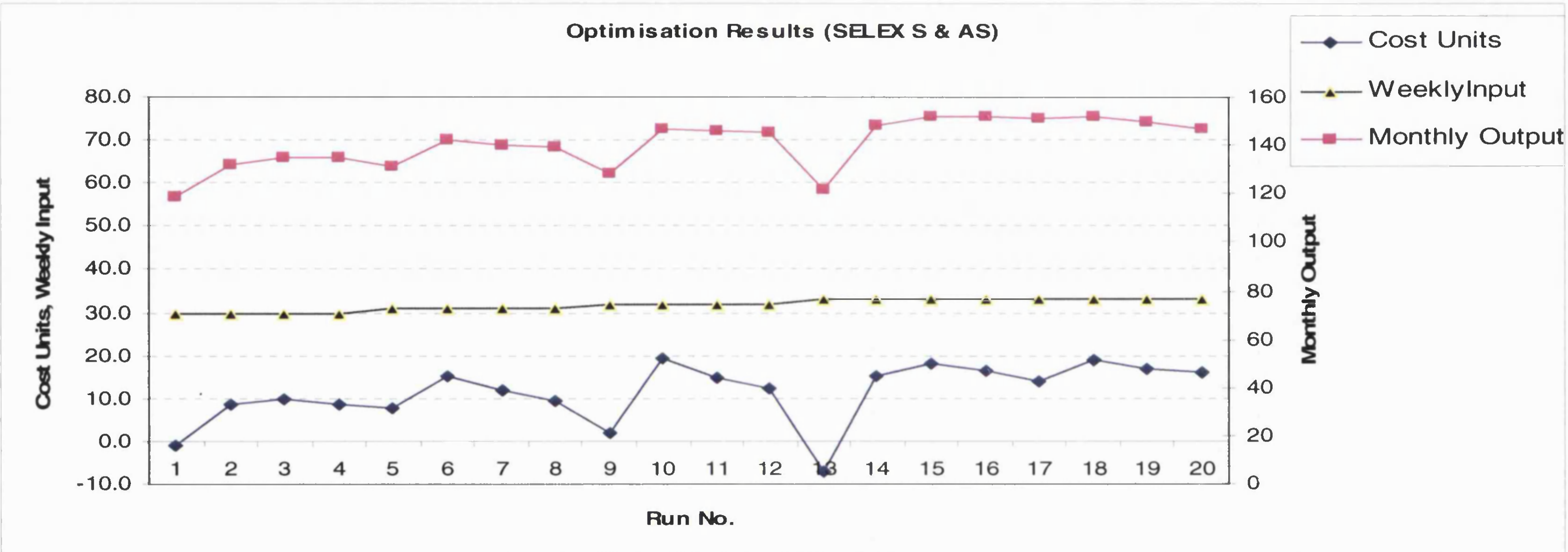


Figure 10.5.2 Optimisation Results (SELEX S & AS) from Expert Mechanism

Sequence No.	Element Name	Qty	Utilisation	Setup	Down Repair	Blocked	Idle	Util & Set-up	Element type
75	AssemblyWeekend	7	81.95154				18.04845	81.95154	5
65	Test	3	71.43619				28.56380	71.43619	5
49	Assembly	7	43.16955				56.83044	43.16955	5
34	OP0081	1	35.27210	0	0	0	10.88966	35.27210	2
28	OP0035	1	32.65306	0	0	0	16.32142	32.65306	2
39	OP0085	1	32.26190	0	0	0	19.17091	32.26190	2
31	OP0055	1	32.04506	0	0	0	14.70727	32.04506	2
26	OP0031	1	31.02891	0	0	4.198979	23.50446	31.02891	2
29	OP0038	1	27.55102	0	0	0	25.94812	27.55102	2
41	OP0110	1	27.47023	0	0	0	21.83035	27.47023	2
30	OP0051	1	27.37244	0	0	8.391156	20.08290	27.37244	2
24	OP0015	1	25.85459	0	0	0	39.24596	25.85459	2
115	Oven2	1	23.33333	0.625	0	0	55.54247	23.95833	2
111	Oven	1	22.29166	0.694444	0	0	53.70717	22.98611	2
35	Q1_Assy	1	22.56377	0	0	0	41.32546	22.56377	2
2	Q1_R	1	22.44897	0	0	0	45.57079	22.44897	2
40	OP0100	1	20.66326	0	0	0	41.88775	20.66326	2
117	Storeman	1	20.57142				79.42857	20.57142	5
119	Kitting	1	20.57142	0	0	0	79.42857	20.57142	2
12	OP0011	1	20.32844	0	0	0	54.29528	20.32844	2
33	OP0071	1	17.98469	0	0	0	41.91581	17.98469	2
44	OP0160	1	15.30612	0	0	0	59.05187	15.30612	2
4	Q2	1	15.15306	0	0	0	65.57823	15.15306	2
32	OP0061	1	14.22619	0	0	0	69.08822	14.22619	2
23	Q9	1	13.46938	0	0	0	64.71152	13.46938	2
42	OP0130	1	11.85185	0	0	0	64.30902	11.85185	2
15	Q3	1	10.10204	0	0	0	79.54825	10.10204	2
25	OP0020	1	9.825680	0	0	0	74.05761	9.825680	2

Table 10.5.3 Part of the Simulation outputs (SELEX S & AS Systems model)

CHAPTER 11
DISCUSSION

11.1 Introduction

The thesis attempts to substantiate the research aims to 1) create an effective technique that could assist production engineers and managers in interpreting results, and in the assessment and automatic optimisation of performance when using manufacturing simulation. 2) derive a method to check model output data consistency for effective optimisation processes. These research aims were developed after studying current practice in industry, and investigation into current research work in manufacturing simulation and performance analyses and enhancement, from both an academic and industrial point of view.

Although a great volume of research has been done on how to make manufacturing simulation easier to use and faster to get results, no significant effort has been made to improve the overwhelming demand at the back-end where a huge amount of results (in a form of tables and graphs) is pushed out for someone to make sense of it. A simulation can be thought of as a mechanism that turns input parameters into output performance measures (Law and Kelton, 2000). In this sense a simulation is just a function, which may be vector valued or stochastic, the explicit form of which is also unknown. The assertion that "simulation packages cannot optimise a system's performance, nor can they solve problems, they can only describe the results of "what-if" questions." still holds with many simulation packages on the market.

Only limited alternative model solutions can be dealt with in a traditional way. As the possible alternatives increase, conducting a large number of simulation runs becomes time consuming and costly. Therefore, simulation modelling becomes a trial-and-error process in which a set of input factors is used to generate output performance measures. The repetitive nature of this approach is often inefficient in terms of time and computing cost as well as in interpretation and prediction of results. Heizer and Render (2005) also support the fact that the trial-and-error approach in simulation does not generate an optimum solution. Cochran and Lin (1992), described the pitfall of computer simulation as producing only ad hoc results in the form of discrete data tables in which system parameters are not explicitly related.

11.2 Simulation Optimisation

The optimisation work of this research adds to a number of publications and commercial applications on simulation optimisation techniques. The relevant authors include Fu and Nelson (2003) and Azadivar (1992) who discuss the integration of simulation and optimisation method; Fu (2002) and April et al (2003) who identify and discuss classical approaches for optimising simulations; Spall (2003) whose book covers an in depth look into stochastic search and optimisation; Andradottir (1998) who discusses the shortcomings of the sample path optimisation method as applied to simulation; Nicolai et al (2004) who looks into the use of response surface methodology as a computer simulation optimisation method; Alam et al (2002), Kilmer et al (1993), and Pierreval and Huntsinger (1992) who proposed simulation meta-models as methods of optimising performance in simulation; Laguna and Marti (2002), and Van Bears (2003) who discuss the use of neural network in meta-models for simulation optimisation; Lyu and Gunasekaran (1997), and Addallah (1994) who discussed the integration of an expert system with simulation to evaluate and improve scheduling strategies; Russel et al (1994) who worked on the use of artificial intelligence to benefit simulation; Brusha and Franek (2003) who proposed the use of metaheuristic methodologies for simulation optimisation; Lanner Group (2004), Luke (2003), Debuse et al (1999), Barretto et al (1999), Kilkpatrick et al (1983), and Moraga (2004) who published work in using simulated annealing, as part of a metaheuristic optimisation method, for simulation optimisation; Glover et al (1999) who discussed tabu search as a metaheuristic approach for solution search strategy; Laguna (1997) who described scatter search as the methodology for simulation optimisation as applied in the commercial optimiser OptQuest; and Pichitlamken and Nelson (2003) who proposed a hybrid system of nested partition and hill climbing for simulation optimisation.

The majority of the above publications and associated commercial application packages work on systematic search from a pool of possible model settings. A more preferred approach would be to employ the practical expertise-oriented methods that manufacturing engineers and managers would generally use (such as the Theory of Constraints and line balancing) that in most instances lead to improved settings with a high level of confidence.

The most commercially applied optimisation methods are based on metaheuristics, predominantly evolutionary algorithms, that perform some sort of search for optimal values of input parameters. Yet, the procedure usually needs to run a large number of alternative models before deciding on the best solution, and it often requires a substantial amount of computer power and time. Referring to the use of AutoStat as an optimiser, Britton (2000) said "Optimisation analysis takes a large number of runs. You can use AutoStat to make runs on multiple machines on your network... You can take advantage of other machines to make runs overnight or on weekends". This indicates that a lot of computer time is wasted on running settings that do not lead to performance improvement.

Fu (2002) reviewed five commercially applied simulation optimisation systems, with the goal of using their routines to seek improved settings of user-selected system parameters with respect to the performance measure(s) of interest. He concluded that contrary to the use of mathematical programming software packages, the user has no way of knowing if an optimum has actually been reached. He added, the biggest problem with currently implemented optimisation methods is that though they may be intelligent in performing the search procedures, they are somewhat oblivious to the stochastic nature of the underlying system. Thus, they completely lack any sense of how to efficiently use a simulation package. It is surprising that RSM, the well established regression method which is quite general and easy to implement has not yet been incorporated into any of the commercial packages.

Additionally, most of the optimisation methods discussed above have not been properly applied in industries due to their requirement of high level of sophistication on the part of the user. April et al (2003) expressed the view that the analytical techniques (such as response surface methodology) require considerable technical depth such that they have not been favoured by industrial users.

11.3 Expert Mechanism

The expert mechanism developed in this research receives outputs from the simulator, assesses system performance, and based on operational rules embedded in it, effects changes to the input variables to enhance performance in the subsequent simulation run. Unlike most of the simulation optimisation systems in the literature that are predominantly based on metaheuristic search methods, this optimiser is based on proven performance enhancing operations management methods such as the Theory of Constraints and line balancing (capacity and flow type). This approach avoids the need to run inferior settings and in the majority of cases leads to improved performance in each iterative run. It also has the advantage that most engineering operations managers and manufacturing engineers have some familiarity to the peformance enhancing methods thus adding confidence into its industrial application. This is supported by Glover and Westwig (2001) who expressed their concern about problems that may arise from the lack of understanding of managers on how integrated simulation-optimization systems work.

The expert mechanism approach has been shown to be more efficient in terms of solution finding time, computer requirements and performance value when compared to one of the most popular commercial simulation optimisers, the Witness Optimiser (see chapter 10 for more detail).

The expert mechanism was integrated to the Witness simulation package through object linked embedding (OLE). Witness is an OLE server which can be controlled by Visual basic which is an OLE controller. The rule-based expert mechanism was written in Visual basic and can thus control the Witness environment leading to a suitable integration of the two. Input data (state variables), are displayed in Excel but controlled by Visual basic. The simulator uses these input data, runs the model and generates simulated results. The expert mechanism receives the relevant output data from the simulator, rearranges the data (with some help from Excel), assesses model performance and generates recommended changes on the next set of state variables used as input data to the simulator. The iteration goes on until a limiting factor is reached.

11.4 Data Consistency Check

The author has not come across any detailed published work that looks into appropriate data consistency checks on simulation model results. Since most simulation models use random variables as input, the simulation output data are themselves random and can be properly assessed by conducting simulation runs.

It has been established that, in a performance improvement sense, the system's variability has to be minimised to an acceptable level before effective performance improvement process can commence. This assertion is supported by Deming (2000) and Robinson (2004) who describe a system with high variability as undesirable and practically difficult to assign resources to the levels of demand. Moreover, effective performance improvement can only be achieved when the output variability has been minimised to an acceptable level. Therefore, it was essential that a data consistency check was conducted especially on models with data of a stochastic nature which are very common in manufacturing systems.

A mechanism using a cusum chart (BS5703 part 3, 2003) has been developed to check simulation output data consistency. The cusum chart is a recognised statistical process control chart known for its ability to detect small changes in performance and its clear graphical representation. Cusum charts are also most advantageous when dealing with a population considered to be homogeneous and a sample of one could be acceptably taken (Oakland 1996, Walker 2005).

Using parabolic V-masks, the cusum chart checks the slope of the graph to determine whether the data are acceptable. If the graph is within the arms of the masks, then the variability of the data is considered as statistically under control and gives a go ahead to the optimisation process.

11.5 Case Studies

Three case studies were carried out to demonstrate the research approach and to consolidate validity of the findings.

Case study one was based on a valve factory model obtained from the Lanner Group, vendors of the Witness simulation package. The case study's main objective was to improve throughput as justified by the constraints of investment. A set of possible investment costs were assigned to the changes in element capacity, speed and quantity. The optimisation process changed these parameters based on its assessment of the model performance from simulation outputs, run the next simulation model, and then assessed the performance improvement against investment. The case study demonstrated that the expert mechanism which is integrated as a controller to the Witness simulator and Excel has managed to effectively improve performance. Throughput was improved in each sequential iterative run showing that the system worked as expected. The toward optimum value showed an improvement in throughput of 332 per cent and a net saving of 36 normalised part units.

The base model was also checked for its output data consistency using the cusum chart. The cusum chart showed that the graph produced from the data under different random number stream settings proved to be within the arms of the upper and lower masks, indicating acceptable statistical consistency.

It could be said that the case study model was limited in many aspects but validated the basic concept of using an expert mechanism to optimise performance in a manufacturing simulation.

Case study two was based on a flywheel and starter ring manufacturing company whose objective was to increase throughput by up to 75 per cent based on two shift working. The resources of the company were limited to the existing system; therefore, the extra demand was to be met through overtime work. The maximum possible overtime work was set for 40 hours.

The expert mechanism used rules based on bypass balancing and the theory of constraints within the existing shop floor system. The operating costs were also assessed at each iterative stage. The case study demonstrated that the expert mechanism was effectively used to automatically enhance the performance of the manufacturing system. The results of the case study showed that the throughput was increased by 70 per cent and the operating cost was reduced by 21 per cent. It should be understood in this case study that the effect of higher work in progress inventory was assumed insignificant.

The output data consistency of the base (initial) model and the last (optimised) model were automatically checked using the cusum chart. The graphs produced from the output data of both models lay within the arms of the upper and lower masks, indicating statistically acceptable consistency.

Case study three was based on a current and real model from a world class company, SELEX Sensors and Airborne Systems (S&AS), Basildon, Essex. The case study's main objective was to validate the optimisation system of this project with real manufacturing data and to compare its strengths and weaknesses with a prominent commercial optimiser. The simulation model had been built, verified and validated by the company. The expert mechanism was to be 1) tested for its ability to control the model, extract relevant data from the model, assess the required performance, make the changes on the input variables (consistent with the constraints) and conduct the next iterative run, 2) compared against a well known commercial simulation optimiser (Witness Optimiser) for its effectiveness and justification of approach.

The SELEX S & AS model was developed initially to support the production setup and to verify the monthly production output and allocation of labour resources. The challenge was to increase production output with minimum resources and to maximise profit.

The expert mechanism automatically extracted the model elements and corresponding input variables from the model and displayed them in Excel. Using the input variables, it run the base model, extracted the simulation outputs, assessed the performance, identified the elements where there are weaknesses, and altered the relevant input variables to enhance performance. The model was then re-run with the new variable values and the process continued until the optimisation process was stopped by the model constraints. The near optimum model obtained through this optimisation process was then subjected to a cost reduction routine where under-utilised resources were tested, and where applicable, their quantity adjusted. The results showed that throughput could be improved by 25 per cent with a justifiable saving of 19.5 normalised profit units.

The results of the expert mechanism were then compared with the commercial optimiser - Witness Optimiser from Lanner Group. The results showed that the expert mechanism was more efficient in converging to the near optimum solution. The main superiority of the expert mechanism was seen in the optimisation run time where the expert mechanism took under 10 percent of the time needed by the Witness Optimiser. Additionally, the expert mechanism was run in a computer that was less powerful than the one used by the Witness Optimiser.

SELEX S & AS consider that the case study has shown that, using optimisation techniques, a more conclusive and more effective approach could be reached in identifying near-optimum production setups that would otherwise take a considerable amount of time or may not be achieved using the traditional "what-if" approach.

The results have shown that this type of optimisation process is practical and could be used to optimise any manufacturing system

11.6 Summary Evaluation of Expert Mechanism and Consistency Checking Element

The expert mechanism, as applied to the three case studies proved that it is an effective tool to enhance performance in terms of the specified measures of performance and in iteration times. The case studies demonstrated that the effectiveness of the Expert mechanism in performance optimisation was markedly visible, and the use of cusum chart for consistency checking was plausible.

The results of case study one satisfied the objective requirements of improving throughput within the constraints of investment justification. The near optimum value obtained using the expert mechanism showed an improvement in throughput of 332 per cent and a net saving of 36 normalised part units (details in Chapter 8). In case study two, the objective was to improve throughput by up to 75 percent without investing any money on hardware. The results of using the expert mechanism resulted in an improvement of the throughput by 70 per cent and in a reduction of a unit cost by 21 per cent (details in chapter 9).

The results of case study three proved that, when compared against *Witness Optimiser* which is one of the widely used simulation optimisation tools, the *expert mechanism* performed better especially in the time needed to converge to the near optimum solution (details in Chapter 10). The expert mechanism took under 10 percent of the time needed by the Witness Optimiser to converge to the near optimum solution.

The case studies also proved that the use of a cusum chart for simulation output data consistency check is plausible. The ability of cusum chart to pick up small variations in data has justified the choice of cusum chart as the tool for data consistency checking. Its application has been successfully illustrated in two of the case studies.

The output data consistency of the base (initial) models and the last (optimised) models in case studies one and two were automatically checked using the cusum chart. The graphs produced from the output data of both models lay within the arms of the upper and lower masks, indicating statistically acceptable consistency (details in chapters 8 & 9).

11.7 Limitations of the Research

One of the main limitations in the research has been that the simulation package, Witness, is proprietary and manipulation of its internal features is limited. An example is the inability to have different run times to effect overtime work on individual work stations. This shortcoming has been compensated to a large extent by adding proportional correction factors to the influencing input variables.

The case studies were limited to manufacturing systems of a steady state and flow type environment. Ideally, the principles of the optimisation element of the research could equally apply to non-steady manufacturing environments but applying the data consistency check may need more in-depth testing when applied to non-steady state conditions.

In all the case studies the effect of work in progress (WIP) inventory has been ignored. When working with expensive products, the effect of WIP inventory may have significant influence on the cost. As such, the use of bypass line balancing has to be properly evaluated.

Although the expert mechanism has a generic nature and has been tested on a real manufacturing problem, more knowledge needs to be added to its knowledgebase to be able to optimise in a wide range of problems.

CHAPTER 12
CONCLUSIONS AND FURTHER WORK

12.1 Introduction

The thesis attempts to substantiate the research proposal that an effective technique of performance optimisation could be coupled with manufacturing simulation to interpret simulation outputs, assess performance, and effect changes to model input variables for the purpose of performance enhancement. It also attempts to substantiate the second proposal that a method can be derived to check simulation model data consistency because high output variability should be avoided in order to conduct an effective performance optimisation process. The proposals were developed after an investigation into the current methods of performance analyses using manufacturing simulation, the approaches in simulation optimisation methods and the lack of simulation output variability checks, both from the academic and commercial points of view.

12.2 Conclusions

12.2.1 Findings of review of the current work

The findings of review of the current work include

- That in most simulation packages a great deal of investigation has been done on how to make manufacturing simulation easier to use and faster to get results but no significant effort has been made to improve the overwhelming demand at the back-end where a huge amount of results is pushed out for someone to make sense of it;

- That most manufacturing simulation packages are designed to work on the "what if" approach and that the repetitive nature of this approach is often inefficient in terms of time and computing cost as well as in the interpretation and prediction of results.

- That the traditional optimisation methods such as gradient search, random search, ranking and selection, and sample path optimisation are mainly based on deterministic systems that their application to complex manufacturing systems with stochastic elements has been very limited.

- That the majority of the commercially applied simulation optimisation methods have been based on metaheuristics predominantly evolutionary algorithms. Their approach is mainly based on systematic search for a solution from a pool of possible alternatives. The procedures usually require a large number of alternative models before deciding on the best solution thus often requiring a substantial amount of computer power and time.

- That the majority of the optimisation methods have not been applied in industries because they require some level of sophistication on the part of the user. In most cases the users are oblivious of the techniques involved in the optimisation process.

- That the author has not come across any detailed published work that looks into model data consistency check on model results.

12.2.2 Contributions to the Body of Knowledge

The main contributions to the body of knowledge can be categorised into two main domains:

- Developing and evaluating an expert mechanism that is integrated with a manufacturing simulator for manufacturing analysis and optimisation. The expert mechanism follows an approach that fills some of the gaps described in section 12.2.1.

- Introducing the cusum chart for checking the data consistency of simulation results where stochastic variability is involved.

12.2.2.1 Expert Mechanism

The main merits and features of the expert mechanism include

- Its knowledge base includes proven manufacturing performance enhancing methods such as the theory of constraints (TOC) and bypass line balancing. These methods are familiar to manufacturing engineers and managers thus add confidence in industrial application.
- The rule-based system in the expert mechanism results in an improved model after each run thus reducing the processing time and cost unlike the current metaheuristic orientated commercial optimisation packages.
- It controls the simulation package (Witness) through the seamless integration of the expert mechanism with the Witness simulation package using object linked embedding (OLE).
- A practical experimentation using a model from a world class company, SELEX S & AS, showed that the expert mechanism could be more efficient in terms of solution finding and computer load requirements when compared with one of the most popular commercial simulation optimisers – Witness Optimiser.
- Three case studies validated the effectiveness of the expert mechanism.

12.2.2.2 Data Consistency Check with Cusum Chart

The important features of the cusum data consistency check include

- Most manufacturing simulation models use random variables and this leads to the simulation outputs themselves being stochastic. High variability in simulation output is not desirable. Additionally, for effective optimisation the variability of the output data should be within acceptable limits (Deming 2000, Robinson

2004D). Cusum checks that the output data are within statistically acceptable limits.

- Cusum is a recognised statistical process control chart well known for its ability to detect small changes in performance; it also has a clear graphical representation.

- In this research the parabolic V-masks are used to check whether the cusum chart which makes use of the slope of the simulation output graph, is within the upper and lower masks.

- If the cusum chart shows unacceptable variability in the simulation output data then the output variability of the system should be minimised (such as by using sensitivity analysis) before optimisation could proceed. This needs testing in future work.

- The case studies demonstrated effectiveness of the use of cusum chart. It may need to be tested further in a real world environment though.

- The cusum chart could also be used to check the statistical level of performance improvement during optimisation.

12.3 Further Work

The areas of further work could be categorised in three main sections:

- Improving and extending the functionality of the integrated system;
- Widening the application areas of the integrated process
- Experimenting with hybrid optimisation methods: expert mechanism and metaheuristic approach.

12.3.1 Improving and Extending the Functionality of the Integrated System

The expert mechanism and the cusum data consistency checker developed in the research, although generic in nature, have limitations in application. This offers a potential for expansion and enhancement.

The expert mechanism-Witness combination used in the research was limited to steady-state, discrete and flow-type manufacturing systems. These systems generally cover the majority of line and batch process manufacturing but many manufacturing systems have transient elements. Production lines that require complete cleaning every evening such as in manufacturing of printed circuit boards are examples. To be applied in these types of problems, the knowledgebase of the expert mechanism needs to be enriched. The generic application of the theory of constrains, line balancing and the like could be used as models for devising rules for generic application of transient elements as well.

Cusum was applied for checking data consistency in this research. Time series and the Welch methods were used to determine the warm-up period in the research case studies. The use of cusum chart for automatic determination of the warm-up period could be explored in future work. The cusum method, as a well known control chart for determining small changes, could identify the section where the variability of the output data is insignificant. This decision making can be incorporated into the expert mechanism thus generating the warm-up period automatically.

Due to Witness being proprietary software, the degree of manipulation of the model is limited. The expert system mainly worked with the input variables and the simulation outputs to assess performance and effect changes in the input data to improve performance. Some possible performance enhancing changes (such as changes in part routing) could not easily be included in an automatic optimisation system. That is, the model flow routing and the resources the parts seize, although their quantity could vary, are considered as nonchangeable. Future work could look into working with the simulation package vendors to attain more flexibility in adjustable variables.

12.3.2 Widening the application areas

One of the areas not considered in the integrated system is quality. The effect of quality and its implications to throughput and cost could be included in the system for a more realistic and comprehensive outcome. Witness has the capability of carrying out a Six-Sigma analysis. Six-Sigma works based on the amounts of defects per million. The application of the expert mechanism could be widened by exploiting this capability of Witness.

12.3.3 Hybrid Optimisation Method

The expert mechanism, using a different approach, tried to fill some of the gaps left open by existing simulation optimisation methods. This does not imply that the expert mechanism cannot benefit by utilising some strong elements of existing optimisation methods. An example could be the use of evolutionary optimisation method where the vicinities of the best solution of the expert system could be explored in search of a better solution. Therefore, a look into applicable metaheuristic methods that can support or enhance the optimisation system could be useful.

REFERENCES

Abdallah M H, 1994, Knowledge-based Simulation Model for Job shop Scheduling, International Journal of Operations Research and Production Management, 15 (10), pp 89-102

Agarwal G and Babu S, 1994, A Simulation Study of Performance of MRP based Systems under Interaction Effects, Production Planning and Control, 5(1), pp 30-46

Alam, F M, McNaught K R, and Ringrose T J, 2002, Investigating Appropriate Experimental Designs for Neural Network Simulation Metamodels, Operations Research Simulation Study Group - Workshop Proceedings, Birmingham, 20-21 march (Operational Research Society), pp.45-54.

Anderson, P G, 2001, Tutorial on Genetic Algorithms, World Manufacturing Congress 2001, Rochester Institute of Technology, US

Andradottir S, 1998, A Review of Simulation Optimisation Techniques, Proceedings of 1998 Winter Simulation Conference, December 13-16, 1998, Washington DC, USA, pp151-158

April, J, Glover, F, Kelly, J P, Laguna, M., 2003, Practical Introduction to Simulation Optimization, Proceedings of the 2003 Winter Simulation Conference, 7-10 December, New Orleans, Louisiana, USA, pp 71-78.

Avello, E A, Baesler F F, Moraga R J, A, 2004, Metaheuristic Based Simulation Annealing for Solving Multiple-Objective Problems in Simulation Optimisation, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp 508-513

Azadivar, F, 1992, Tutorial on Simulation Optimisation , Proceedings of the 1992 Winter Simulation Conference, Proceedings of the 24th Winter Simulation Conference, December 13-16, Arlington, VA, USA, pp.198-204

Baccouche M, Boukachour J, and Benabdelhafid A, 2003, Genetic Algorithm for Scheduling Aircraft Landing, , Proceedings of Industrial Simulation Conference 2003, Valencia, Spain, Eurosis Publication, pp.418-422

Baesler F F, Moraga M, Ramis F J, 2002, Productivity Improvement in the Wood Industry using Simulation and Artificial Intelligence, Proceedings of 2002 Winter Simulation Conference, San Diego, California, 8-11 December, USA, pp 1095-1098

Balci O, 1994, Validation, verification, and testing techniques throughout the life cycle of a simulation study, Annals of Operations Research, 53, pp 121-173.

Barretto, M R P, Chwif L, Eldabi T, Paul J R, 1999, Simulation Optimisation with the Linear move and Exchange move Optimisation Algorithm, Proceedings of the 1999 Winter Simulation Conference, Phoenix, Arizona, 4-8 December, USA, pp. 806-811.

Baudin M, Mehorta V, Tullis B, Yeaman D, Hughes R A , 1992, From Spreadsheet to simulations, 4th International Semiconductor Manufacturing Science Symposium (IEEE), June 15-16, 1992, San Francisco, CA, USA, pp 94-99

Boesel J, Bowden R O, Glover F, Kelly, J P, Westwig E, 2001, Future of Simulation Optimisation, Proceedings of the 2001 Winter Simulation Conference, December 9-12, 2001, Arlington, VA, USA, pp 1466-1469

Box, G E P, and Draper, N R, 1987, Emprical Model Building and Response Surfaces, John Wiley & Sons

Bitron, 2000, Optimizing AutoMod models with Autostat, Autoflash monthly newsletter, Autosimulations.

Browne, Harhen, Shivnan, 1988, Production Management Systems – A CIM Perspective, Addison Wesley Publishing Company

Brusha & Franek, 2003, Inducing Parameters of decision tree for expert system shell MCESE by Genetic Algorithm, Proceedings of Industrial Simulation Conference (ISC2003), Valencia, Spain, Eurosis publication, pp.48-52

Bryman A, 1989, Research Methods and Organisational Studies, Unwin Hyman: London, pp 230-233

Burrell G and Morgan G, 1979, Sociological Paradigms and Organisational Analysis, Heineman: London, pp 2-9

BS 2564, Standards on control chart techniques, British Standards Institute, 1955

BS 5703, 2003, Data Analysis and quality control using cusum techniques, Part 3, British Standards Institute

Carry, A, 1998, Simulation of Manufacturing Systems, John Wiley & Sons.

Carson J S, 1992, Modelling, Proceedings of the 23rd Winter Simulation Conference, Phoenix, Arizona, USA, December 8-11, pp82-87

Cheng R C H and Currie C S M, 2004, Optimization by Simulation Metamodelling Method, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp485-489

Clarke P, 1972, Action Research and Organisational change, Harper & Row: London, pp 6-25

Cleric M, 1999, The swarm and the queen: towards a deterministic and adaptive particle swarm optimisation, Proceedings of the Congress on Evolutionary Computation, Washington DC (Piscatoway:IEEE Service Center), pp 1951-1957.

Cleric M, 2002, The particle swarm: explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, 6, pp 58-73

Cochran K J, Lin L, 1992, Assembly Line System Dynamic Behaviour for High Priority Job Order Processing, International Journal of Production Research, 30(11), Taylor and Francis Ltd, pp 1683-1697

Coleman, 1996, Quality Assurance in Transnational Education, Journal of Studies in International Education, Vol. 7, No. 4, pp 354-378

Debuse, J C W, Rayward-Smith, V J, Smith, J.D., 1999, Parameter Optimisation for a Discrete Event Simulation, Journal of Computers and Industrial Engineering, 37, 181-184.

Deming W, 2000, Out of the Crisis, CUP Publishing NY, pp 309-369

Dengiz B and Alabas C, 2000, Simulation Optimisation using Tabu Search, Proceedings of the 2000 Winter Simulation Conference , WSC 2000, 10-13 December, Orlando, FL, USA, pp 805-810

Dorigo M, and Stutzle T, 2002, The ant colony optimisation metaheuristic: Algorithm applications and advances, Glover and Kochenberger editors, Handbook of Metaheuristics, Kluwer Academic Publishers.

Dorigo M, Maniezzo V, 1996, The ant system: optimisation by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics, 26(1), pp 1-13

Fernihough A M, 1997, An Integrated Technique for the Evaluation of Business Strategies, PhD Thesis, University of Bath, pp 58-62

Friedman, Pressman, 1988, The Meta-model in Simulation Analysis: Can It be Trusted?, Journal of Operational Research Society, Volume 39, pp 939-948

Fu M and Nelson B, Guest Editorial on Simulation Optimisation, ACM Transactions on Modelling and Computer Simulation, 13(2), ACM NY, 2003, pp 105-107

Fu M C, Andradottir S, Carson S C, Glover F, Harrell C R, Ho Y C, Kelly J P, Robinson S M, 2000, "Integrating Optimisation and Simulation: Research and Practice", Proceedings of the 2000 Winter Simulation Conference , WSC 2000, 10-13 December 2000, Orlando, FL, USA. pp 610-617

Fu M C, 2002, Optimisation for Simulation: Theory and Practice, INFORMS Journal of computing, 14 (3), pp 192-215.

Glover F and Laguna M, 1997, Tabu Search, Kluwer Academic Publishers, Boston

Glover F, Kelly J P, Laguna M, 1999, New Advances for Wedding Optimisation and Simulation, Proceedings of the 1999 Winter Simulation Conference, 4-8 December, Phoenix Arizona, USA,pp 255-260

Goldratt and Cox, 2004, The Goal: A Process of Ongoing Improvement, North River Press, Inc. NY

Hans-Peter W, 1993, Production Planning and Control – The tool to Ensure Logical Quality, Robotics and Computer Integrated Manufacturing, 10(1/2), pp 99-107

He D W and Kusiak A, 1997, Design of assembly system for modular products, IEEE Transactions, Robotics and Automation (13), pp 646-655

He Z, and Wei C, 1999, A new population-based incremental learning method for the travelling salesman problem, Congress on Evolutionary Computation, Washington DC (Piscatoway:IEEE Service Center), pp 1152-1156.

Heizer J and Render B, 2005 A, Operations Management, Pearson Prentice Hill, pp195-204

Heizer J and Render B, 2005 B, Operations Management, Pearson Prentice Hill, pp583-593

Heizer J and Render B, 2005 C, Operations Management, Pearson Prentice Hill, pp 450-456

Homem-De-Mello, 2003, Variable Sample Methods for Stochastic Optimisation, ACM Transactions on Modelling and Computer Simulation, 13 (2), pp 108-133.

Howard K, and Peters J, 1990, Managing Management Research, Management Decision, 28(5), MCB University Press

Hurrion, R D, 1986, Simulation – International Trends in Manufacturing Technology, IFS Publication Ltd

ISO 8258 (BS 7785), Shewhart control charts, British Standards Institute, 1991

Jones, M H, White K P, 2004, Stochastic Approximation with Simulated Annealing as an Approach to Global Discrete Event Simulation Optimisation, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp 500-506

Joustra P E and Van Dijk N M, 2001, Simulation of Check-in at Airports, Proceedings of the 2001 Winter Simulation Conference, WSC 2001, December 9-12, 2001, Arlington, VA, USA, pp1023-1028

Kalpakjian S, 1995, Manufacturing Engineering and Technology, 3rd Edition, Addison-Wesley, pp1-2

Kelton D W D, Sadowski R P, Sturrock D T, 2004A, Simulation with ARENA, McGraw Hill, pp 11-12

Kelton D W D, Sadowski R P, Sturrock D T, 2004B, Simulation with ARENA, McGraw Hill, pp 42-43

Kennedy J and Spears W M, 1998, Matching Algorithms to problems: An experimental test of the particles swarm and some genetic algorithms on the multimodal problem generator, Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, Alaska (Piscataway: IEEE Service), pp 78-83.

Kennedy J, Eberhart R, 1995, Particle Swarm Optimisation, Proceedings of the IEEE international conference on Neural Networks, Piscataway NJ, pp 1942-1948

Khorami M, 1992, Application of Rule-based Systems to Computer Integrated Manufacturing, Proceedings of the 12th Annual Conference of the British Computer Society, 16-17 December, Cambridge, UK, pp 1-19

Kilmer R A & Smith A E, 1993, Using Artificial Neural Networks to Approximate a Discrete Event Stochastic Simulation Model, Intelligent Engineering Systems through Artificial Neural Networks, 3, Eds Dagli,Burke,Fernandez & Gosh, ASME Press, pp 631-636

Kilmer, R A, Smith, A E, Shuman, L.J., 1997, An Emergency Department Simulation and a Neural Network Metamodel, Journal of the Society for Health Systems, 5, 63-79.

Kirkpatrick S, Gelatt, C D, Vecchi M P, 1983, Optimisation by Simulated Annealing, Science, 220 (4598), 671-680

Kleijnen J, 1979, Regression Metamodels for Generalising Simulation Results, IEEE Transactions on System, Man, and Cybernetics, Volume SMC-9, No 2. (pp 93-96)

Kuhn T S, 1962A, The Structure of Scientific Revolutions, University of Chicago Press: Chicago, pp10

Kuhn T S, 1962B, The Structure of Scientific Revolutions, University of Chicago Press: Chicago, pp103

Laguna M, Marti R, 2002, Neural Network prediction in a system for Optimisation, IIE Transactions, 34 (3), pp 273-282

Laguna M, 1997, Optimisation of Complex Systems with OptQuest, University of Colorado, Boulder, White paper April 8,1997, http://www.decisioneering.com/optquest/complexsystems.html (cited 5 January 2005)

Lanner Group, 1998, Witness Training Material, Lanner Group, U.K.

Lanner Group, 2004, OLE Automation in Witness, Witness 2004 on-line, Lanner Group.

Law, M A, & Kelton, D.W., 2000 (A), Simulation Modelling and Analysis (London: McGraw Hill), pp1-6

Law, M A, & Kelton, D.W., 2000 (B), Simulation Modelling and Analysis (London: McGraw Hill), pp 669-671

Law, M A, & Kelton, D W, 2000 (C), Simulation Modelling and Analysis (London: McGraw Hill), pp 83-86

Law, M A, & Kelton, D W, 2000 (D), Simulation Modelling and Analysis (London: McGraw Hill), pp 662-665

Law, M A, & Kelton, D W, 2000 (E), Simulation Modelling and Analysis (London: McGraw Hill), pp 268-279

Liu B, Hussein A A, McKay B, 2004, Classification rule discovery with ant colony optimisation, IEEE Computational Intelligence Bulletin, February 3(1), pp 31-35.

Luke B T, 2003, Simulated Annealing, Learning FromTheWeb.net , (White Paper) http://members.aol.com/btluke/simann1.htm (cited 12 September 2004)

Lyu & Gunasekaran, 1997, An intelligent Simulation Model to Evaluate Scheduling Strategies in a steel Company, International Journal of Systems Science, 28(6), 611-616.

Maier N R F, Psychology in Industry, 1965, 3rd edition, Houghton-Mifflin, Boston, pp 417-419.

Maniezzo V, Gambardella L M, Luigi F, 2004, Ant colony optimisation, www.cs.unibo.it/bison/publications/aco2004.pdf (cited 7 May 2006)

ME50173, 2006, Research Methodology, Department of Mechanical Engineering, University of Bath, http://www.bath.ac.uk/mech-eng/units/me50173 (cited 15 May 2006)

Micheleti G F, 1987, Simulation in Manufacturing, Proceedings of the 3rd International Conference, SIM3, Turin, Italy.

Miller I, Freund J E, Johnson R A, Probability and statistics for engineers, 3rd Edition, Prentice-Hill NJ, 1990

Montgomery D C, Introduction to statistical quality control, 4th Edition, Wiley NY, 2001

Morito S, Koida J, Iwama T, Sato M, Tamura Y, 1999, Simulation-based constraint generation with applications to optimisation of logistic system design, Proceedings of the 1999 Winter Simulation Conference, Edited by Farrington, Nembhard, Sturrock & Evans, www.informs-cs.org/wsc99papers/074.pdf

Nicolai R P, Dekker R, Piersma N, Van-Oortmarssen G J, 2004, Automated Response Surface Methodology for Stochastic Optimisation Models with Unknown Variance, Proceedings of the 2004 Winter Simulation Conference, December 5-8, Washington DC, USA, pp 491-499

Noreen E, Smith D, Mackey J T, 1995, The theory of constraints, The North Rivers Press

Oakland J S, 1986, Statistical process control, Heinemann London

Onwubolu G C, Clerc M, 2004, Optimal path for automated drilling operations by a new heuristic approach using swarm optimisation, International Journal of Production Research, 42(3), pp 473-491

Owen M, 1989, SPC and continuous improvement, Bedford:IFS.

Pegden C D, Shannon R E, Sadowsky R P, 1995, Introduction to Simulation using SIMAN, McGraw Hill: NY.

Petie D E, 1992, Computing through Manufacturing, 18 (5), Journal of the Institute of Operations Management (IOM), pp25-27.

Philips E M and Pugh D S, 1987, How to get a Ph.D., Milton Keynes, Open University, pp 45

Pichitlamken J and Nelson B L, 2003, A Combined Procedure for Optimisation via Simulation, ACM Transactions on Modelling and Computer Simulation, vol 13(2), ACM NY, pp 155-179

Pierreval, and Huntsinger, 1992, An investigation on Neural Network Capabilities as Simulation Metamodels, Proceedings of the 1992 Summer Computer Simulation Conference, San Diego, California, USA (SCS), pp.413-417.

Randel L, Holst L, Bolmsjo G, 1999, Concurrent Development of Large Discrete-Event Simulation Models, Proceedings of World Manufacturing Congress, September, Durham, UK, ICSC Academic Press, pp 77-83.

Robinson S, 2004 (A), Simulation: The Practice of Model Development and Use, John Wiley, pp 2-12

Robinson S, 2004 (C), Simulation: The Practice of Model Development and Use, John Wiley, pp 51-55

Robinson S, 2004 (B), Simulation: The Practice of Model Development and Use, John Wiley, pp68 -70

Robinson S, 2004 (D), Simulation: The Practice of Model Development and Use, John Wiley, pp 176-177

Robinson S, 2004 (E), Simulation: The Practice of Model Development and Use, John Wiley, pp 194-195

Robinson S, 1994, Successful Simulation, McGraw Hill

Roli A, 2002, AIRO news editorial, 7(3) Autumn, http://www.sci.unich.it/~aroli/pubs/roli_aco-aironews.pdf (cited 11 January 2006)

Rose G, 1982, Deciphering Sociological Research, Macmillan: London, pp14-18

Russell, 1994, An Approach to Standardising the Simulation to Knowledge System Interface, Society for Computer Simulation, 63 (3) pp143-158

Russel & Taylor, 2006, Operations Management, John Wiley and Sons, pp 222-232

Saad and Byrne, 1995, Simulation Metamodelling and its Application to a Flexible Hybrid Assembly System, Advances in Manufacturing Technology IX, Proceedings of the Eleventh National Conference on Manufacturing Research, Taylor and Francis Publishers, pp413-417

Shi L, Olafsson S, 2000, Nested partitions method for stochastic optimisation, Meth, Comput. Appl. Prob., 2, 271-291

Slack N, Chambers S, Johnston R, Operations Management, Prentice Hall, 2004, pp111-116

Spall, J C, 2003, Introduction to stochastic search and optimization, John Wiley & Sons.

Swain J, 2006, Simulation Software Survey, Real World Operations Research, http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html (Accessed on 12/10/2006)

Systems Modelling Corporation, 1992, The Future of Simulation Software, ARENA Tutorial pack.

Hill T, 2004, Production Operations Management, Prentice Hall, pp 53-77,
Thomasma & Li-xiang, 1991, Object-Oriented Simulation for Manufacturing Information Handling System Design, Simulation Series , The Society for Manufacturing Simulation (SCS) pp118- 122

Tiwari M K, Prakash, Kumar A, Mileham A R, 2005, Determination of an optimal assembly sequence using the psychological algorithm, IMechE Journal of Engineering Manufacture, 219(B), pp 137-149

Trafford V, 2001, An Unravelling of Research Methodology, Earlybrave Publications Ltd, pp 7-12

Ulgen and Thomasma, 1990, SmartSim: An Object Oriented Simulation Program Generator for Manufacturing Systems, International Journal of Production Research Vol 28, The Institute of Production Engineers, Taylor and Francis Ltd., pp 1713-1730

Van Beers W C M, Kleijnen J P C, 2003, Kriging for interpolation in random simulation, Journal of the Operations Research Society, 54 (3), pp 255-262

Van Dijik N M, 2000, On Hybrid Combination of Queuing and Simulation, Proceedings of the 2000 Winter Simulation Conference , WSC 2000, 10-13 December 2000, Orlando, FL, USA, pp 147-150

Vonderembse M A, and White G P, 2004 A, Operations Management, John Wiley, pp173-181

Vonderembse M A, and White G P, 2004 B, Operations Management, John Wiley, pp 391-399

Vonderembse M A, and White G P, 2004 C, Operations Management, John Wiley, pp 247-258

Walker R W, 2001, Evolutionary Operation (EVOP) Evaluated by Cusum charts, Conference paper at World Manufacturing Congress 2001, ISBN 3-906454-28-2

Walker R W, 2005, Statistical Process Control, Teaching Notes, Anglia Ruskin University

Walton R and Gaffney M, 1991, Innovating to Complete: Lessons for Diffusing and Managing Change in the Work Place, Jossey-Bass

Waterman D A, 1986, Guide to Expert Systems, Addison_Wesley

Webster's Ninth New Collegiate Dictionary, 1983, Merriam-Webster Inc

Wu B, 1994 (A), Manufacturing Systems Design and Analysis, Chapman & Hall, pp 4-5

Wu B, 1994 (B), Manufacturing Systems Design and Analysis, Chapman & Hall, pp 215-220

Zhang W, Xie X, Bi D, 2004, Handling boundary constraints for numerical optimisation by particle swarm flying periodic search space, Congress on Evolutionary Computation (CEC), Oregon, USA, pp 2307-2311.

APPENDIX A: LIST OF PUBLISHED PAPERS

- Mebrahtu H, Walker R W, Mileham T., 2004, Performance Enhancement Using an Expert Mechanism in a Manufacturing Simulator. IMechE. 218 (Part B): Feb 2004, pp 245-249,

- Mebrahtu H, Walker R W, Mileham T, 2004, Knowledge-based Optimisation in a Manufacturing Simulator, 21st International Manufacturing Conference, Limerick, Ireland, Gemini International Ltd., pp 243-250.

- Walker R W, Mebrahtu H., 2004, Automating part of the model verification process. Industrial Simulation Conference 2004, Eurosis Publications, pp 384-388.

- Mebrahtu H, Walker R W, Mileham T., 2003, Performance Enhancement Using an Expert Mechanism in a Manufacturing Simulator, Industrial Simulation Conference 2003, Valencia, Spain, Eurosis Publications, pp 299-303

- Mebrahtu H, Walker R W, 2002, Manufacturing Simulation and Analysis. The Fifth SMESME International Conference, Chelmsford, U.K, Earlybrave Publications Ltd., pp202-208.

- Mebrahtu H, 2002, Automatic Analysis of Manufacturing Simulation Results, OR Simulation Study Group Workshop, Birmingham, U.K, The Operations Research Society, pp145-150

- Mebrahtu H., 2001, Prediction of Shop Floor Performance Using A Flexible Manufacturing Simulator, World Manufacturing Congress 2001, Rochester, U.S.A., ICSC-NAISO Academic Press. 1913-032: 57 (CD)

APPENDIX B: CODE LISTING - CASE STUDY TWO

Option Explicit

```
 Public MODELNAME As String

Dim gobjWitness As Object
```

Public Sub RunWitnessModel()

```
   On Error GoTo RunWitnessModel_Err

   ChDrive (Left(ThisWorkbook.Path, 1))
   ChDir (ThisWorkbook.Path)
   If OpenWitness() Then
      If LoadWitnessModel(MODELNAME) Then
         TransferDataToWitness
         RunWitness Sheets("control").Cells(4, 2), True
         'GetWitnessResults
         GetWitnessResults "FinalwheelP", "finalwheelD"
         Modelchanges (85)

         CloseWitness False, False

      End If
   End If

RunWitnessModel_Exit:
   Exit Sub

RunWitnessModel_Err:
   MsgBox Error$
   Resume Next
End Sub
```

Public Sub ConsTest_Ini()

```
   On Error GoTo RunWitnessModel_Err
   Dim S As Integer, RS As String, flywheelP As String, flywheelD As String

   ChDrive (Left(ThisWorkbook.Path, 1))
   ChDir (ThisWorkbook.Path)
   Worksheets("ConsTest").Range("C3:AA5").ClearContents  'Clear cusum cells
    If OpenWitness() Then
     For S = 0 To 24
```

```
    If LoadWitnessModel(MODELNAME) Then
      TransferDataToWitness
      RS = "RStream"
      gobjWitness.variable(RS) = S
      RunWitness Sheets("control").Cells(4, 2), True
      flywheelP = "FinalwheelP"
      flywheelD = "FinalwheelD"
      Sheets("ConsTest").Cells(3, S + 3) = S + 1
      Sheets("ConsTest").Cells(4, S + 3) = gobjWitness.Function("nship", flywheelP)
      Sheets("ConsTest").Cells(5, S + 3) = gobjWitness.Function("nship", flywheelD)

    End If
    Next S

      'CloseWitness False, False

  End If

RunWitnessModel_Exit:
  Exit Sub

RunWitnessModel_Err:
  MsgBox Error$
  Resume Next
End Sub
```

Public Sub ConsTest_Last()

```
  On Error GoTo RunWitnessModel_Err
 Dim S As Integer, RS As String, flywheelP As String, flywheelD As String

  ChDrive (Left(ThisWorkbook.Path, 1))
  ChDir (ThisWorkbook.Path)
  If OpenWitness() Then

    Worksheets("ConsTest").Range("C41:AA43").ClearContents
    'Worksheets("Control").Range("F13:AJ27").ClearContents
   For S = 0 To 24
    If LoadWitnessModel(MODELNAME) Then
      TransferDataToWitness
      RS = "RStream"
      gobjWitness.variable(RS) = S
      RunWitness Sheets("control").Cells(4, 2), True
      flywheelP = "FinalwheelP"
      flywheelD = "FinalwheelD"
      Sheets("ConsTest").Cells(41, S + 3) = S + 1
      Sheets("ConsTest").Cells(42, S + 3) = gobjWitness.Function("nship", flywheelP)
```

```
        Sheets("ConsTest").Cells(43, S + 3) = gobjWitness.Function("nship", flywheelD)
      End If
      Next S

        'CloseWitness False, False

    End If

RunWitnessModel_Exit:
    Exit Sub

RunWitnessModel_Err:
    MsgBox Error$
    Resume Next
End Sub
```

Public Function OpenWitness() As Boolean

```
On Error GoTo OpenWitness_ValidToContinue
    Dim nRetVal As Integer
    Dim counter As Integer
    Dim sWITNESS As String

    sWITNESS = Sheets("control").Cells(2, 2) + "\WITNESS.EXE" ' the WITNESS
command line
    MODELNAME = Sheets("control").Cells(3, 2)    ' the model to use
    OpenWitness = True 'assume it will succeed

    ' Detect whether Witness is currently open
    gobjWitness = GetObject(, "Witness.WCL")

    'successful connection - Witness is already open.  Exit now
    OpenWitness = True

Exit Function

OpenWitness_ValidToContinue:
    'clear the active error
    Resume OpenWitness_Continue

OpenWitness_Continue:
    On Error GoTo OpenWitness_WaitingToLoad

    ' Witness is not active - it is valid to continue

    ' Kick off WITNESS - sWitness is Witness path & exe
```

```
    Shell sWITNESS, vbMinimizedFocus

   ' Give WITNESS time to load
OpenWitness_WaitingToLoadResume:
   DoEvents
   ' Connect to WITNESS object
   Set gobjWitness = GetObject(, "Witness.WCL")

On Error GoTo 0
   ' Wait for open sequence to complete
   Do
   Loop While gobjWitness.ModelStatus() = 3

   OpenWitness = True
   Exit Function

OpenWitness_WaitingToLoad:
   Resume OpenWitness_WaitingToLoadResume
End Function
```

Public Function LoadWitnessModel(strModel As String) As Boolean

```
   On Error GoTo NoObject
   gobjWitness.BeginOLE

   On Error GoTo LoadWitness_Error
   gobjWitness.wcl ("LOAD " & strModel)

   Do
   Loop While gobjWitness.ModelStatus = 3

   gobjWitness.EndOLE
   LoadWitnessModel = True

   Exit Function

LoadWitness_Error:
   gobjWitness.EndOLE

NoObject:
   MsgBox ("Error loading " & strModel)
   LoadWitnessModel = False
End Function

Sub TransferDataToWitness()
```

```
Dim h As Integer, g As Integer
Dim Exp As String, element As String, X As String


  On Error GoTo GetWitnessResults_Err
' Activate Witness application
   AppActivate ("WITNESS")
 Do While gobjWitness.ModelStatus <> 2
   DoEvents
  Loop
'-------------- Loading data from VB -------------------
For h = 1 To gobjWitness.Count
   g = h + 10
   element = gobjWitness.Name(h)
   Exp = "CycT_or_Cap"
   If gobjWitness.Type(element) = 2 Then
     gobjWitness.variable(Exp, h) = Worksheets("Control").Cells(g, 4).Value
   End If
' --------- A way of entering capacity of conveyors and buffers! ------------

   If gobjWitness.Type(element) = 4 Then
    ' Exp = "CycT_or_Cap(" & h & ")"
      Exp = "set capacity " & element & " " & Worksheets("Control").Cells(g, 4).Value
     gobjWitness.Action (Exp)
   End If

   If gobjWitness.Type(element) = 3 Then
   X = Chr$(13) & Chr$(10)
   Exp = "DETAIL" & X & "SELECT" & X & element & X & "PART LENGTH:" &
Worksheets("Control").Cells(g, 4).Value
   Exp = Exp & ";" & X & "END " & element & X & "END SELECT" & X & "END
DETAIL"
   gobjWitness.wcl (Exp)
    End If

' -------------------------------------------------------------------------

Next h

GetWitnessResults_Err:
' error handling

End Sub

Sub RunWitness(strStop As String, bBatch As Boolean)
On Error GoTo RunWitness_Error
```

```
  If bBatch Then
    gobjWitness.Batch (strStop)
  Else
    gobjWitness.Run (strStop)
  End If

  Do While gobjWitness.ModelStatus = 2 And gobjWitness.variable("TIME") = 0
    DoEvents
  Loop

  Do While gobjWitness.ModelStatus <> 2
    DoEvents
  Loop
  Exit Sub
RunWitness_Error:
  If bBatch Then
    MsgBox ("Error batching Witness to " & strStop)
  Else
    MsgBox ("Error running Witness to " & strStop)
  End If
End Sub
```

Sub GetWitnessResults (part1 As String, part2 As String)

```
Dim i As Integer, k As Integer, shipped As Integer
Dim element As String, intpress As String, strCopy As String, strSort As String
Dim intN As Integer, intP As Integer, Run_num As Integer, run_cell As Integer


   On Error GoTo GetWitnessResults_Err
' Activate Witness application
    AppActivate ("WITNESS")
 Do While gobjWitness.ModelStatus <> 2
    DoEvents
  Loop
  ' Record number of parts shipped, average WIP & number of run

Sheets("control").Cells(5, 5).Value = Sheets("control").Cells(5, 5).Value + 1
run_cell = Sheets("control").Cells(5, 5).Value + 4
Sheets("control").Cells(7, run_cell).Value = Sheets("control").Cells(5, 5)
Sheets("control").Cells(8, run_cell) = gobjWitness.Function("nship", part1) +
gobjWitness.Function("nship", part2)
Sheets("SHEET1").Cells(2, 2) = gobjWitness.Function("nship", part1) +
gobjWitness.Function("nship", part2)
Sheets("sheet1").Cells(2, 5) = gobjWitness.Function("awip", part1) +
gobjWitness.Function("awip", part2)
```

```
'--------------------- Extra message box! -------------------------------
'shipped = gobjWitness.Function("nship", part)
'intpress = MsgBox("The total finished parts shipped are " & shipped, vbOKOnly)
'-------------------------------------------------------------------------

For i = 1 To gobjWitness.Count
   k = 3 + i
   element = gobjWitness.Name(i)
   Sheets("SHEET1").Cells(k, 1) = i
   Sheets("SHEET1").Cells(k, 2) = element
   Sheets("SHEET1").Cells(k, 3) = gobjWitness.quantity(element)

   If gobjWitness.Function("Putil", element, 2) > 100 Then
   Sheets("SHEET1").Cells(k, 4) = ""
   Else
   Sheets("SHEET1").Cells(k, 4) = gobjWitness.Function("Putil", element, 2)
   End If

   If gobjWitness.Function("Putil", element, 4) > 100 Then
   Sheets("SHEET1").Cells(k, 5) = ""
   Else
   Sheets("SHEET1").Cells(k, 5) = gobjWitness.Function("Putil", element, 4)
   End If

   If gobjWitness.Function("Putil", element, 5) > 100 Then
   Sheets("SHEET1").Cells(k, 6) = ""
   Else
   Sheets("SHEET1").Cells(k, 6) = gobjWitness.Function("Putil", element, 5)
   End If

   If gobjWitness.Function("Putil", element, 3) > 100 Then
   Sheets("SHEET1").Cells(k, 7) = ""
   Else
   Sheets("SHEET1").Cells(k, 7) = gobjWitness.Function("Putil", element, 3)
   End If

   If gobjWitness.Function("Putil", element, 1) > 100 Then
   Sheets("SHEET1").Cells(k, 8) = ""
   Else
   Sheets("SHEET1").Cells(k, 8) = gobjWitness.Function("Putil", element, 1)
   End If

   If gobjWitness.Function("Putil", element, 2) + gobjWitness.Function("Putil", element,
4) > 100 Then
   Sheets("SHEET1").Cells(k, 9) = ""
```

```
    Else
    Sheets("SHEET1").Cells(k, 9) = gobjWitness.Function("Putil", element, 2) + _
        gobjWitness.Function("Putil", element, 4)
    End If
    Sheets("SHEET1").Cells(k, 10) = gobjWitness.Type(element)

Next i

'=============== Sorting Data =================

intN = gobjWitness.Count + 3
intP = 2 * gobjWitness.Count + 6
strCopy = "J" & intN
strSort = "J" & intP

'Worksheets("Sheet1").Range("B3:I20").Copy

Worksheets("Sheet1").Range("A3:" & strCopy).Copy _
    Destination:=Worksheets("Sorted_data").Range("A3")

Worksheets("Sorted_data").Range("A4:" & strSort).Sort _
    Key1:=Worksheets("Sorted_data").Range("I5"), order1:=xlDescending, _
    Key2:=Worksheets("Sorted_data").Range("D5"), order2:=xlDescending

'==============================================

GetWitnessResults_Err:
' error handling
End Sub
```

Sub Modelchanges (Limit As Variant)

```
Dim j As Integer, m As Integer, ElemNo As Integer, Pre_run As Integer
Dim strEType As String, intpress As String, CycT As String, element As String
Dim No_of_Changes(1 To 20) As Integer, N As Integer, y As Integer, run_cell As Integer

'Columns 24 to 27 indicate the 4 most busy elements;
'Busy for more than 90% is considered to need action, but
'for less than 90%, increasing the buffers is enough.

run_cell = Sheets("control").Cells(5, 5).Value + 4
If run_cell >= 6 Then
  Worksheets("Control").Range(Cells(11, run_cell - 1), Cells(27, run_cell - 1)).Copy
        'Worksheets("Control").Range(Cells(11, run_cell), Cells(23, run_cell)).Paste
```

```
  ActiveSheet.Paste Destination:=Worksheets("Control").Range(Cells(11, run_cell),
    Cells(27, run_cell))
  'Destination:=Worksheets("Control").Range(Cells(11, run_cell), Cells(23, run_cell))
End If

For j = 4 To 30
 ElemNo = Worksheets("Sorted_data").Cells(j, 1).Value
  m = 10 + ElemNo
  element = gobjWitness.Name(ElemNo)
  'Pre_Elem = ElemNo - 1
 If Worksheets("Sorted_data").Cells(j, 9).Value > Limit Then
   If Worksheets("Sorted_data").Cells(j, 10) = 2 Then    'If the element is a machine,
      If No_of_Changes(ElemNo) < 21 Then              'The # of 2-hour overtime changes
        No_of_Changes(ElemNo) = No_of_Changes(ElemNo) + 1
        'CycT = "CycT_or_Cap(" & ElemNo & ")"
         'Worksheets ("Control").Cells(m, 4) = 0.9 * Worksheets("Control").Cells(m, 4).
              Value
        y = No_of_Changes(ElemNo)
        Worksheets("Control").Cells(m, 4).Value = (40 / (40 + 2 * y)) *
              Worksheets("Control").Cells(m, 4).Value
            'This is the modified process time to compensate for the 2- hour overtime work
               '40 hours is the basic one-shift work time

        'Sheets("control").Cells(5, 5).Value = Sheets("control").Cells(5, 5).Value + 1

         If run_cell >= 6 Then

           Pre_run = run_cell - 1
          Worksheets("Control").Cells(m, run_cell).Value = Worksheets("Control").Cells(m,
                Pre_run).Value + 2
         End If

      Else
        intpress = MsgBox("Maximum overtime of 40 hours reached! ")
      End If
   Else
     If Worksheets("Sorted_data").Cells(j, 10) = 3 Then    'If the element is a conveyor

    Worksheets("UsedElement").Cells(1, 1) = gobjWitness.Used(element)


'============== Identifying the preceding Element here!========

    '   For N = 24 To 24 + 40
     '    If Worksheets("Sheet1").Cells(N, 1).Value = Worksheets("Sheet1").Cells(j,
1).Value - 1 Then
```

```
    '    If Worksheets("Sheet1").Cells(N, 7).Value > 25 Then
    '      Worksheets("Control").Cells(m, 2) = Worksheets("Control").Cells(m, 2).Value
+ 1
       '  End If
        '  Exit For
         ' 'Else
           ' intpress = MsgBox("No Change needed on " & ElemNo & "!", vbOKOnly)
   '      End If
    '   Next N

    ' Else
     '  intpress = MsgBox("Element NOT a machine or a conveyor!", vbOKOnly)
    End If
  'End If
 'Else
  ' If Worksheets("Sheet1").Cells(10, Pre_Elem) = 2 Then
   '  Worksheets("Control").Cells(m, Pre_Elem) = Worksheets("Control").Cells(m,
Pre_Elem).Value + 1
  End If

 End If
 Next j

End Sub
```

Sub Reset()

```
Worksheets("Control").Cells(5, 5) = 0
Worksheets("Control").Range("B11:B27").Copy _
   Destination:=Worksheets("Control").Range("D11")

Worksheets("Control").Range("E7:AJ8").ClearContents
Worksheets("Control").Range("F13:AJ27").ClearContents
'Worksheets("Control").Range("H28:AJ32").ClearContents
End Sub
```

Sub CloseWitness(bSavePrompt As Boolean, Optional bSave As Boolean)

```
 On Error GoTo CloseWitness_Err

   'Activate Witness application
   AppActivate ("WITNESS")

   ' send Alt-F4
   SendKeys "%{F4}"
```

```
    ' Handle save dialog if necessary
    If Not IsMissing(bSave) Then
        If bSave Then
            SendKeys "Y", True
        Else
            SendKeys "N", True
        End If
    End If

CloseWitness_Exit:
    Exit Sub

CloseWitness_Err:
    MsgBox Error$
    Resume Next
 End Sub
```

APPENDIX C: WITNESS OPTIMISER SETTINGS (CASE STUDY 3)

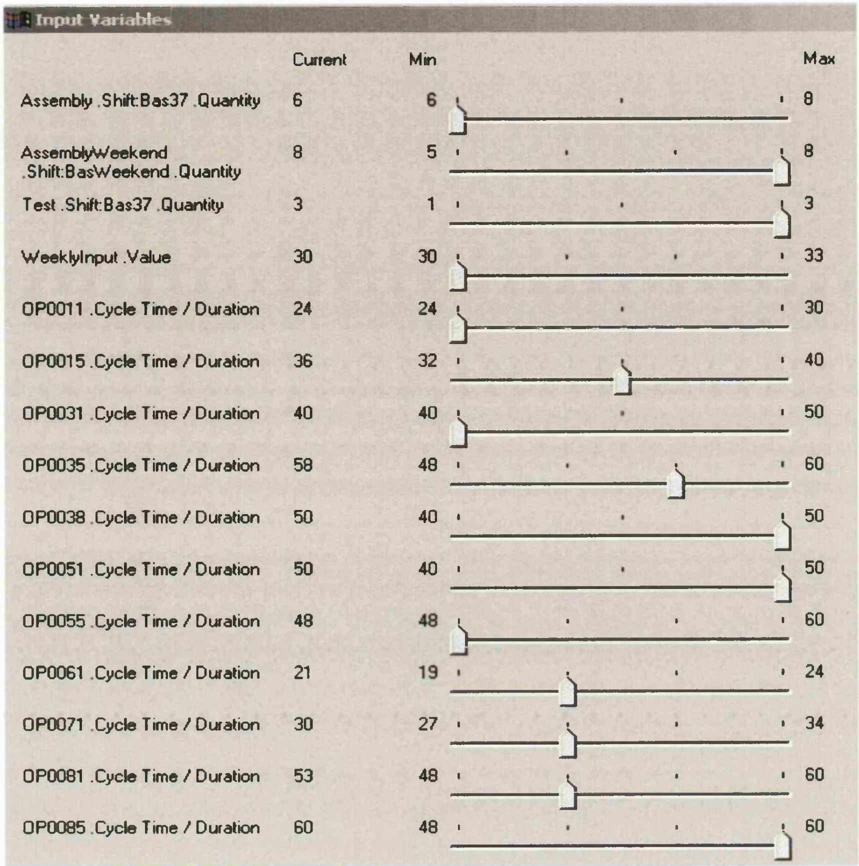


Figure C1. Optimization Variables Settings

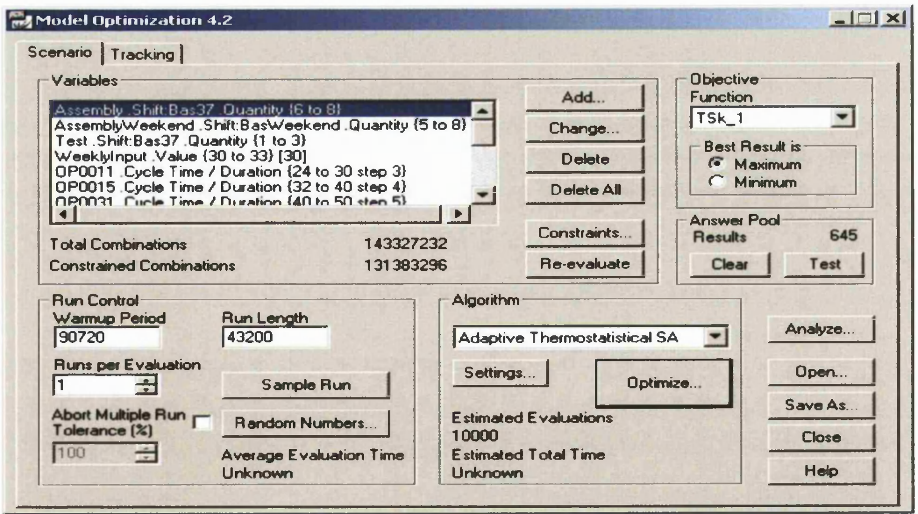


Figure C2. Optimizer model Screen

APPENDIX D: CUSUM CHART MASK DESIGN (BS 5703-3,2003)

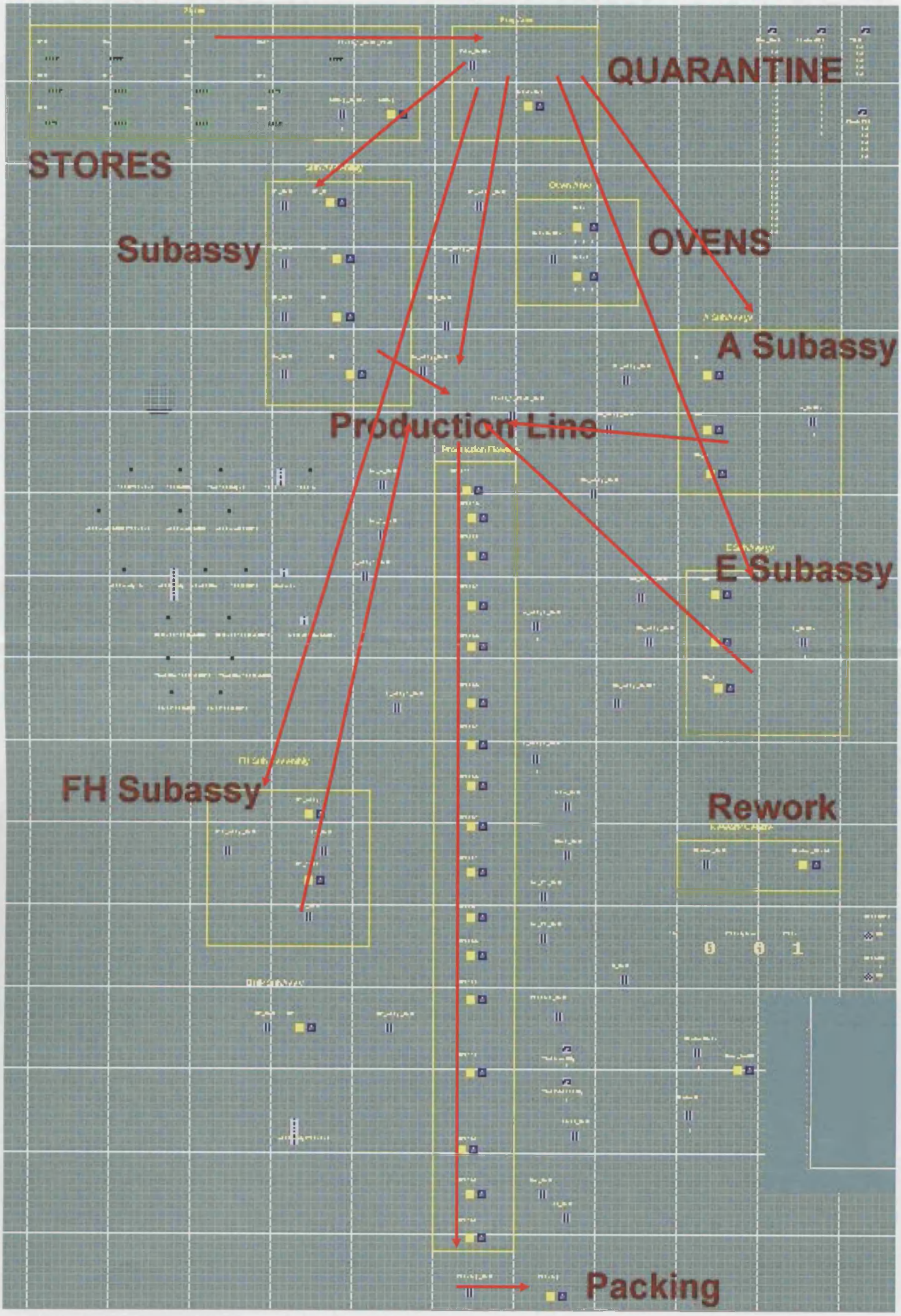
$$\sigma_e \text{ the standard error} = \frac{\sigma}{\sqrt{n}}$$

Individual readings

The C1 mask design is 10s, 5s and 10 parts mask.
The C2 mask design is 8.5s, 3.5s and 10 parts.

Data to construct a semi-parabolic mask

Distance from datum (parts)	Half width C1 mask (units standard error)	Half width C2 mask (units standard error)
0	1.25	1.0
1	3.1	2.3
2	4.65	3.6
3	5.9	4.7
4	6.85	5.5
5	7.5	6.0
6	8.0	6.5
7	8.5	7.0
8	9.0	7.5
9	9.5	8.0
10	10.0	8.5
15	12.5	11
20	15.0	13.5



WITNESS model of SELEX S & AS

APPENDIX F: A SAMPLE PUBLISHED PAPER

Performance enhancement using an expert mechanism in a manufacturing simulator

H Mebrahtu[1]*, R Walker[1] and T Mileham[2]
[1]School of Design and Communication Systems, Anglia Polytechnic University, Chelmsford, Essex, UK
[2]Department of Mechanical Engineering, University of Bath, UK

Abstract: The need to include a mechanism that could assist in analysis and performance enhancement of simulation models has been under discussion for a long time. Many simulation packages on the market offer powerful 'what if' evaluation techniques for production planning. However, most of them rely on the user's own experience to interpret the results after each simulation, and anyone without such experience would find it difficult to make reasonable sense of the results before deciding on the next simulation run. This paper describes the use of an expert mechanism that could be integrated into a simulation package to facilitate the process of interpreting and assessing simulation results and in improving performance. It also discusses the need for checking stability of the model before reporting the model as realistic.

Keywords: manufacturing simulation, optimization, performance analysis

1 INTRODUCTION

Although simulation in manufacturing has traditionally been used for high-level capacity planning, there are many other benefits in using simulation. Factory layout, production routing, production mix and throughput prediction, bottleneck identification, new resources deployment, to name but a few, can all be predicted using simulation. While explaining the need for simulation, Ben-Arieh [1] and Law and Kelton [2] assert that, in the modern manufacturing facility, the available flexibility introduces another degree of flexibility in decision-making. The lack of clear understanding of the dynamics and interaction of components of modern manufacturing systems calls for the use of simulation as an essential support tool. Simulation is no more a niche management tool that can only be afforded by a few, thanks to ever-increasing computer power and its affordable price. The advancement in programming and software engineering also means that very clever simulation software has hit the market, with highly configurable user features and powerful animation [3].

However, these powerful features are generally focused on the front-end of creating a manufacturing model easily and on getting simulation results quickly [4]. As a result, massive reports, which include statistics, tables and a lot of raw data, are generated, but do not help the user see the connection of these reports with the next appropriate action in a consistent and logical way. Any interpretation and action will depend solely on the user's experience in using simulation. Additionally, limited alternative simulation models could be dealt with in a traditional way, but, as the possible alternatives increase, conducting a large number of simulation runs becomes time consuming and costly [5].

Some commercial simulation packages now include some type of integrated optimization routine, Optimizer in Witness and OptQuest in Delma [6], for instance. The goal of these routines is to seek improved settings of user-selected system parameters with respect to the performance measure(s) of interest. However, unlike mathematical programming packages, there is no way of knowing that an overall optimum has actually been reached, and thus *optimization* may be a loose word.

The experimental work in this paper illustrates the use of a rule-based algorithm that is integrated with a simulation package to analyse simulation output, assess performance of a production floor and automatically change the controllable variables within given constraints to enhance performance. Once the stability of the original model is checked, each time the simulation is executed, the rule-based algorithm would interpret and analyse the results, and suggest a suitable action plan for the next iteration for further improving the performance. Such a concept also opens up a huge

The MS was received on 16 September 2003 and was accepted after revision for publication on 13 October 2003.
**Corresponding author: School of Design and Communication Systems, Anglia Polytechnic University, Victoria Road South, Chelmsford, Essex CM1 1LL, UK.*

possibility of running the rule-based simulator remotely across the Internet, hence allowing smaller companies to benefit from simulation.

2 EXPERIMENTAL MODEL

2.1 Base model

In order to demonstrate how simulation results can be translated into action plans, and how different production scenarios can be compared using performance indices, a case study factory model with limited operation and resource flexibility has been set up as shown in Fig. 1. The experiment was based on a company model obtained from the Lanner Group [7], the software house behind the Witness simulation modelling system. Some operational data have been modified for simplicity.

At the start of the experiment, the case study company experienced a severe backlog in sales orders owing to antiquated machinery and poor production planning. Assuming that there was a demand for up to 5 times the current product output, a series of simulation runs was set up to evaluate the effect of investing appropriate resources against the possible increase in throughput and benefits.

The model consists of seven main operations. The manufacturing process starts with the stock of bars coming into the saw area stock buffer. The bars are then cut, producing three blocks from each bar. After sawing, the blocks go to a belt conveyor that transfers the cut bars to the coating operation. The coating machine coats six blocks at a time. Once coated, the blocks are placed in the staging area adjacent to the inspection station. The inspectors then determine the coating quality of each block and send it either to hardening or to the rework buffer. The hardened blocks are then loaded into special fixtures so that four blocks can enter a grinder at once. There are two grinders available, with no priorities between them. Once ground, the fixture and the four blocks are placed into an unloading station where the blocks (now valves) are sent to the finished stock areas and the fixtures onto an overhead conveyor. The conveyor puts the fixtures back into the fixture buffer for reuse by the loading machine. Witness was used to model the system.

2.2 Model stability

It is important to ensure that the model is not significantly affected by changing the random number streams. If this occurs, then the model results cannot be expected to give a solution that would be realistic. The stability of the model was checked by conducting 25 runs with different random number streams for each run and for each element and each corresponding set of data. This was meant to give the feel that events were following

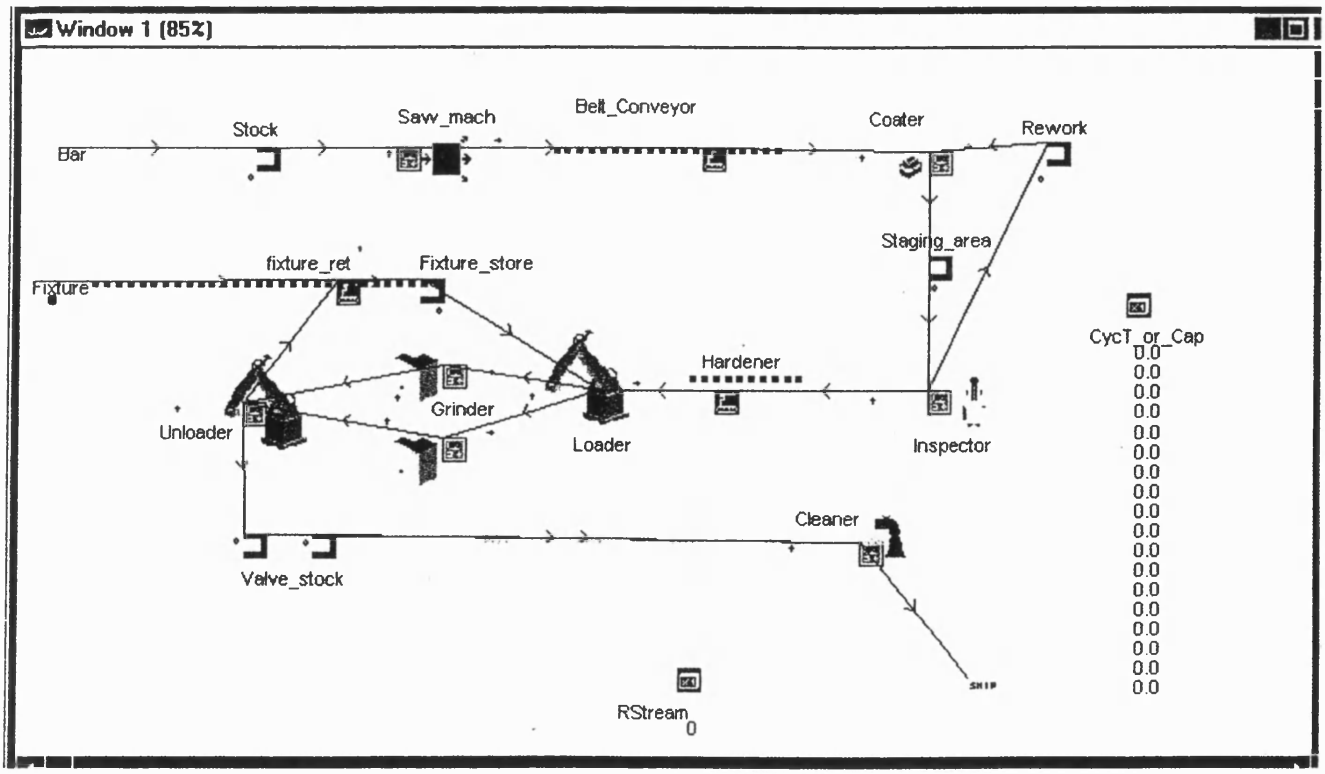


Fig. 1 Simulation model used in the experiment

more realistic randomness. Outputs of the 25 runs were recorded, and a Cusum test (90 per cent confidence) was conducted as shown in Fig. 2. The mask used is a C2 semi-parabolic mask as defined in BS 5703 Part 3 [8]. It could be seen from the graph that the data were all between the upper mask and the lower mask, indicating good consistency. A similar stability check was conducted on the last model, and it again showed satisfactory stability.

The reason for selecting Cusum charts to check stability is to allow future development of the program automatically to check that the change being proposed will result in a significant improvement. The development process will continue by integrating an 'evolutionary operation' (EVOP) design of experiment system into the program, which will be controlled and monitored by Cusum charts [9], until an optimum solution is achieved.

3 RULE-BASED EXPERT MECHANISM

3.1 Objectives

The existing system can manufacture around 144 valves every 75 h. It has been established that the benefits of an increase in throughput by one valve can be fully justified for an investment of £250. That is, for each investment of £250, there must be an increase of at least one part. A maximum amount of £75 000 is available to be spent for the investment, which amounts to an equivalent of 300 more valves to justify the spending. The main investment costs expressed in terms of production benefits are shown in Table 1. Each item has been assigned a cost equivalent in parts.

3.2 Methodology and results

As previously described, the main performance index is net profit (or net saving) which is the difference between the increase in throughput and the investment (expressed in terms of equivalent parts). The main rules used include the techniques of theory of constraints and line balancing, backed by concurrent monitoring of investment. This involves mainly identifying bottlenecks and blockages which are used as the basis for actions to be taken in each sequential simulation run.

Witness as an object link embedding (OLE) automation server could be controlled by Visual Basic (VB) which is an OLE controller [10]. Relevant input/output data to Witness as well as running of Witness could be controlled with VB (with some assistance from Excel). Therefore, using VB to develop the expert mechanism was ideal. The simulator uses data displayed in Excel but controlled by VB, runs the model and generates output. The expert mechanism receives the relevant output data from the simulator, manipulates the data, assesses model performance and generates recommended changes for the next run. The iteration goes on until a limiting factor is reached, at which point the result would be output.

Eleven simulation runs were conducted, with the summary of results shown in Fig. 3. The results indicate that all except runs 4 and 7 could be justified for their respective investments. Models 10 and 8 showed the better net savings, with model 10 significantly favoured both in savings and throughput in terms of rectifying the current problems of the company.

Table 1 Possible costs for investment

	Investment cost equivalent in parts		
Element	New element	Decrease cycle time by 10%	Decrease set-up time by 10%
Buffers	0.2		
Saw machine	100	10	20
Conveyor		10	
Coating machine		24	
Inspector	80	24	
Hardener (furnace)		24	
Loader/unloader	64		
Grinder	200	40	
Cleaner	40		

4 CONCLUSION

Although the model is limited in many respects, it highlighted the basic concept of integrating an optimizing element to a manufacturing simulator for automatic results analysis and performance enhancement. Various performance assessment methods such as throughput, inventory level, machine utilization and investment can be incorporated into future experiments to make the system more versatile for a wider spectrum of simulation scenarios. The proposed concept can handle a mix of different production objectives whereby users can set target figures with each objective, and the system will iterate until those targets are met within specified allowance.

With the ever-growing popularity of the Internet, making the system Web compliant is another goal in future research. When fully developed, registered users from remote sites will be able to use the system by providing required inputs to the simulator, target objectives, constraints of scenarios, plus other necessary details required to build and run a totally customized model on the net. The simulation system will then run continuously at the host website until the targets and constraints are satisfied. The remote user will then be able to view the optimized results and the accompanying conditions. This concept of application on demand (AOD) has yet to be realized but has great potential in allowing smaller firms to benefit from specialized application software

Run No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Valves, x	136	145	144	144	148	148	140	148	144	145	150	152	156	154	140	148	148	152	154	148	148	144	152	147	140
x-T	-11	-2	-3	-3	1	1	-7	1	-3	-2	3	5	9	7	-7	1	1	5	7	1	1	-3	5	0	-7
CuSum	-11	-13	-16	-19	-18	-17	-24	-23	-26	-28	-25	-20	-11	-4	-11	-10	-9	-4	3	4	5	2	7	7	0
Upper mask															41.86	39.4	36.93	34.47	32.01	29.55	27.08	23.14	17.73	11.33	4.924
Lower mask															-41.9	-39.4	-36.9	-34.5	-32	-29.5	-27.1	-23.1	-17.7	-11.3	-4.9

Target, T	147
StdDev	4.924

Ctrl Factors	8.5	8.0	7.5	7.0	6.5	6.0	5.5	4.7	3.6	2.3	1.0

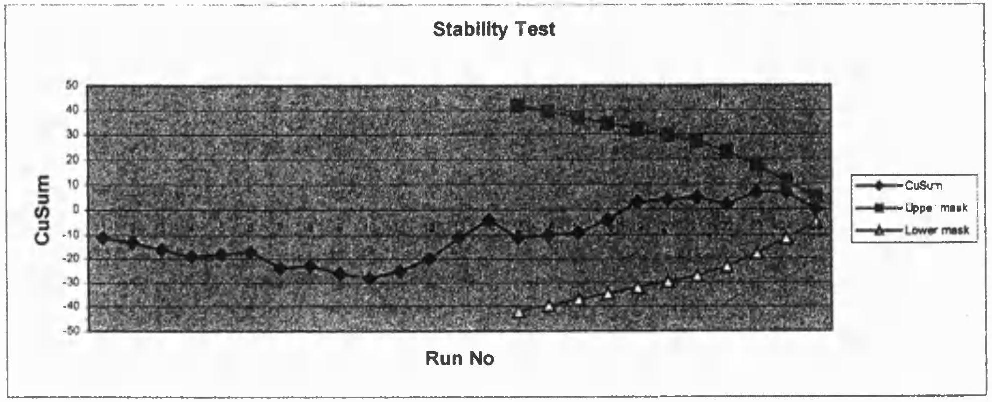
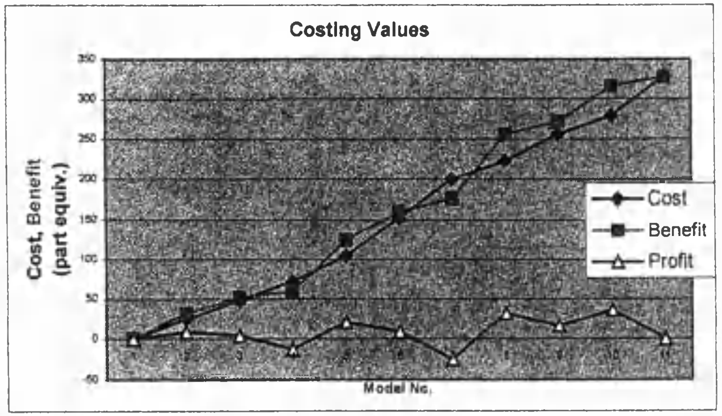


Fig. 2 Stability test (Cusum test) initial model



Model No.	1	2	3	4	5	6	7	8	9	10	11
Valves shipped	136	168	188	196	260	296	312	392	408	452	464
Cost	0	24	48	72	104	152	200	224	256	280	328
Benefit (increase in parts)	0	32	52	60	124	160	176	256	272	316	328
Profit (Saving)	0	8	4	-12	20	8	-24	32	16	36	0

Fig. 3 Simulation results and costing

such as manufacturing simulation, with the consent of the software suppliers.

ACKNOWLEDGEMENTS

Parts of this paper were presented at the Eurosis Industrial Simulation Conference 2003 (ISC'2003)—Eurosis Publication ISBN 90-77381-03-1. Permission to reproduce the paper has been granted by Eurosis.

REFERENCES

1 **Ben-Arieh, D.** *A Knowledge-Based System for Simulation and Control of FMS*, 1986, pp. 287–295 (IFS Publications Limited, UK).
2 **Law, A.** and **Kelton, W. D.** *Simulation Modelling and Analysis*, 1991 (McGraw-Hill, New York).
3 **Mebrahtu, M.** and **Lung, A.** Prediction of shop floor performance using a flexible manufacturing simulator. In World Manufacturing Congress 2001, ICSC-NAISO, 2001 (Academic Press, Canada and The Netherlands).
4 **Pengen, C. D., Shannon, R. E.** and **Sadowski, R. P.** *Introduction to Simulation*, 1990 (McGraw-Hill, New York).
5 **Morito, S., Koida, J., Iwama, T., Sato, M.** and **Tamura, Y.** Simulation-based constraint generation with applications to optimisation of logistic system design. In Proceedings of the 1999 Winter Simulation Conference (Eds R. A. Farrington, H. B. Nembhard, D. T. Sturrock and G. W. Evans), 1999 (Institute of Electrical and Electronic Engineers, Piscataway, New Jersey).
6 **Fu, M. C., Carson, J. S., Harrell, C. R., Kelly, J. P., Andradottir, S., Glover, F., Ho, Y.** and **Robinson, S. M.** Integrating optimisation and simulation: research and practice. In Proceedings of the 2000 Winter Simulation Conference (Eds J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick), Orlando, Florida, 2000, pp. 610–617.
7 Witness training course notes from Lanner Group, 1998.
8 BS 5703: Part 3: 1981 *Data Analysis and Quality Control using Cusum Techniques*, 1981 (British Standards Institution, London).
9 **Walker, R. W.** Evolutionary operation (EVOP) evaluated by Cusum charts. In World Manufacturing Congress, Rochester Institute of Technology, Rochester, New York, 2002.
10 OLE automation in Witness. Witness 2002 on-line, Lanner Group, Redditch, UK, 2002.